THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR ....Garth Churney....

TITLE OF THESIS ....Computation of Synthetic Seismograms....

....by Ray Methods....

....

DEGREE FOR WHICH THESIS WAS PRESENTED ....M.Sc....

YEAR THIS DEGREE GRANTED ....1977....

THE UNIVERSITY OF ALBERTA

COMPUTATION OF SYNTHETIC SEISMOGRAMS BY RAY METHODS

by

© GARTH CHURNEY

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE

DEPARTMENT OF PHYSICS

EDMONTON, ALBERTA

FALL, 1977

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled COMPUTATION OF SYNTHETIC SEISMOGRAMS BY RAY METHODS submitted by Garth Churney in partial fulfilment of the requirements for the degree of Master of Science.

ABSTRACT

Ray methods of generating synthetic seismograms have been favored in seismology over the more exact wave solutions since they may be applied to problems with more complex boundary conditions.

The ray method and code generating scheme of Hron (1972) as it applies to vertical incidence has been reviewed in detail. A new method of generating the ray code based on the concept of permuted partitions has been developed in an attempt to reduce the high computer storage costs associated with the generation scheme used by Hron. Restricting the partitions according to the number of parts forms the basis of attempts to further reduce costs by eliminating uninteresting phases. The effect of including geometric spreading in the amplitude computations has been examined and found to be of significant importance. The ray method of n^{th} order multiples is briefly described and compared with the method of Hron. Finally the complete wave solution for a sixteen layer model is presented and compared with previous results.

ACKNOWLEDGEMENTS

Thanks are due to my supervisor Dr. E.R. Kanasewich for his guidance throughout the project and to Dr. F. Hron for his useful discussions and encouragement.

I would also like to thank Mr. Claude Athias for the many late-hour discussions on topics of current interest, as well as my other associates and friends, Mr. Daniel Au, Dr. Allan Bates, Mr. Dave Ganley, Mr. Jens Havskov, Dr. Roy Hibbs, Mr. Joseph Lee, Mr. K. Sprenke, and Mr. Larry Sydora, all of whom contributed to making my stay more enjoyable.

Special thanks are due to Mr. Dave Ganley for making available his computer program for calculating the complete wave solution to the sixteen-layer model.

In addition I would like to thank Chevron Standard Limited who provided me with support in the form of a graduate fellowship.

TABLE OF CONTENTS

Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE		Page
18	Synthetic Seismograms for Severity Level 3 and Primaries Plus Up to Second Order Multiples..................................	53
19	Comparison of Spike Sequences for N^{th} Order Multiples and N^{th} Order Surface Multiples..................................	56
20	Comparison of Synthetic Seismograms for N^{th} Order Multiples and N^{th} Order Surface Multiples..................................	57
21	Comparison of the Unrestricted and Complete Wave Solutions to the 16-Layer Model..................................	59

CHAPTER I

THE RAY METHOD OF HRON (1972)

1.1 Introduction

A common approach taken in the construction of synthetic seismograms is the ray tracing technique. In this method a group of rays is selected and the amplitude contribution and time of arrival of each ray within the group is determined at a particular location on the surface. The problem lies in the selection of a group of rays from the infinite number of rays travelling from the source to the receiver. There have been various approaches taken to this problem and these will be discussed and comparisons made using synthetic seismograms at vertical incidence.

Practically, synthetic seismograms at vertical incidence are of limited interest however their construction can serve a useful purpose. In exploration work, for example, when the depth of energy penetration is large compared with the shot-geophone spacing, a synthetic seismogram at vertical incidence may approximate to a high degree of accuracy one constructed taking into account the shot-geophone spacing. This may be especially true if the near surface structure has little in the way of velocity contrasts. Of greater importance though is the fact that synthetics at

vertical incidence offer a reasonably inexpensive method of comparing different ray selection criteria. The results of the comparison can then be used as a basis for ray path selection when proceeding to non-vertical modeling.

One approach to the problem of ray selection is due to Hron (1972) with the aid of prior work done in the Soviet Union. The remainder of this chapter consists of a review of Hron's work as it applies to vertical incidence.

1.2 Basic Principles

Rays which travel from the source to receiver along different ray paths with identical travel times are defined as kinematic analogs (Hron, 1972). Within each group of kinematic analogs there may exist subgroups of rays which have equal amplitudes and which are defined as dynamic analogs. The necessary and sufficient condition for kinematic equivalence of two different ray paths or phases is that each must have equal number of segments in each layer. It will be assumed that both the source and receiver are located on the surface, hence the number of segments in each layer must be even. That is, for every down going ray in layer i there must exist a corresponding up going wave. Failure of this condition results in an incomplete ray path.

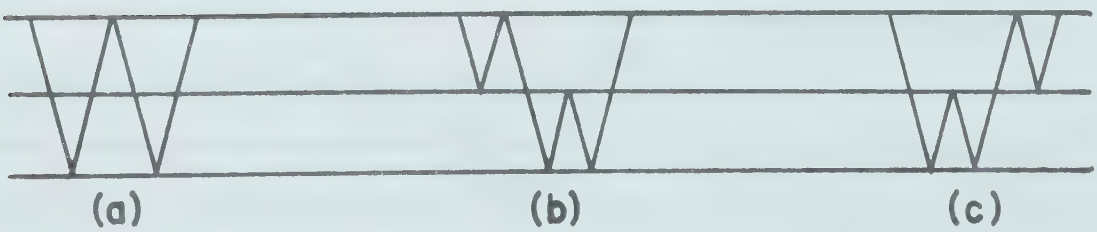
To describe and identify groups of kinematic analogs, J integers are used as the parameters, letting the symbol of the group be

$$(n_1, n_2, \ldots, n_J) \tag{1}$$

where n_j is an integer equal to half the segments in layer j and J is the total number of layers involved.

Figure 1 illustrates the concepts of kinematic and dynamic analogs. The three phases drawn comprise the group of kinematic analogs with symbols (2,2), each ray path drawn having two half-segments in the first layer and two half-segments in the second layer. Within this group of kinematic analogs there exists a subgroup composed of phases (b) and (c) which have the same number and type of interactions with the interfaces and which are called dynamic analogs. The modification of the coding in (1) to include a description of the dynamic properties of a particular phase will be dealt with later.

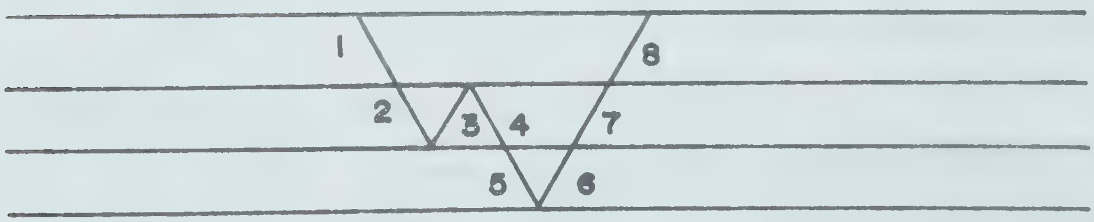
Figure 1. Illustration of the Concepts of Kinematic and Dynamic Analogs.



For a more complete discussion of kinematic and dynamic analogs several important concepts are necessary. A coupled segment is defined as one down going and the nearest up going segment in a particular layer. Since the number of segments in each layer is even there are n_j coupled segments in each layer j. For example in Figure 2 segments 2 and 3 constitute the first coupled segment in layer 2. Segments 4 and 7 constitute the second coupled segment in this layer.

An element of the j^{th} class is defined to be the continuous chain of segments bounded by a coupled segment in the j^{th} layer. Any ray path bounded by a coupled segments in layer 2 for example will be defined as an element of the second class. Hence segments 2 and 3 constitute the first element of the second class and segments 4, 5, 6 and 7 constitute the second element of the second class. Since

Figure 2. Classification of Ray Path Segments



each element of the j^{th} class is bounded by a coupled segment they cannot overlap and since there are n_j coupled segments in the j^{th} layer there are exactly n_j elements of the j^{th} class for all j. If an element of the j^{th} class consists of only two segments such as the first element of the second class consisting of segments 2 and 3 it is called a trivial element. A normal element of the j^{th} class is defined as consisting of at least one coupled segment in the (j+1)st layer. An example would be the second element of the second class consisting of segments 4, 5, 6 and 7.

1.3 Kinematic Analogs

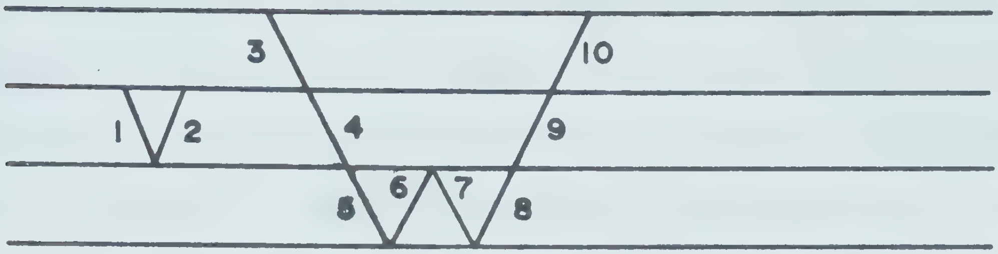
Once the kinematic code is specified thus establishing a particular group of analogs the segments must be connected to form a continuous ray path. Hron (1972) defines continuity in terms of elements of the j^{th} class. He states:

> "The required continuity of the chain of segments is satisfied if at least one element of the j^{th} class $j = 1, J - 1$, is normal so that the segments in the j^{th} and $(j+1)^{th}$ layers are connected. Moreover, all elements of the $(j+1)^{th}$ class within the element of the j^{th} class must be linked together."

These criteria fail to ensure continuity as can be demonstrated by referring to Figure 3.

In Figure 3 there is one element of the first class consisting of segments 3 to 10. The two elements of the second class involve segments 1 and 2, and segments 4 to 9. Finally there are two trivial elements of the third class

Figure 3. Continuity of Ray Paths



consisting of segments 5 and 6 and segments 7 and 8. The element of the first class and the second element of the second class are both normal. Moreover, the element of the second class within the element of the first class and both elements of the third class within the element of the second class are linked together. Despite satisfying Hron's criteria the ray path is not continuous.

In order to ensure continuity at least one element of the j^{th} class, j = 1, J - 1 must be normal. In addition, the (K+1)st coupled segment (if it exists) in layer j must be traceable from the start of the k^{th} coupled segment in the same layer.

To determine the total number of different ray paths, $N_k(n_1,n_2,...,n_J)$ which may be constructed given $2n_j$ segments in each layer j = 1,...,J, one must always start from n_1 elements of the first class which must be linked together. In the case of only one layer (J=1) there is only one way

of connecting the n_1 trivial elements of the first class so that $N_k(n_1) = 1$. If the number of layers is two (J=2) with $2n_1$ segments in the first layer and $2n_2$ segments in the second layer, the total number of different kinematic analogs is given by the number of different ways of distributing the n_2 trivial elements of the second class among the n_1 elements of the first class. An example may serve to illustrate. There are three ways of distributing the two trivial elements of the second class among the two elements of the first class drawn in Figure 4. These three different possible associations are shown in Figure 1(a), (b) and (c). In Figure 1(a) one trivial element has been

Figure 4. Distribution of Trivial Elements into "Empty" Elements of the First Class.

distributed into the first element of the first class; the second trivial element going into the second element. In 1(b) both trivial elements have been distributed into the second element, the first element being left "empty". In

1(c) both trivial elements have gone into the first element and in this case the second element has been left "empty". As can be seen the problem is identical to the distribution of balls into pockets, some of which may be left empty. According to Appendix A the total number of different ray paths possible given $2n_1$ segments in the first layer and $2n_2$ segments in the second layer is

$$N_k(n_1,n_2) = \frac{(n_1 + n_2 - 1)!}{n_2!(n_1 - 1)!} = C_{n_2}^{n_1+n_2-1} \tag{2}$$

For the three-layer case the number of different ways of distributing the n_3 trivial elements of the third class into each arrangement of the n_2 elements of the second class is

$$\frac{(n_2 + n_3 - 1)!}{n_3!(n_2 - 1)!} = C_{n_3}^{n_2+n_3-1} \tag{3}$$

But there are $C_{n_2}^{n_1+n_2-1}$ different distributions of the n_2 elements of the second class among the n_1 elements of the first class. Hence there are $C_{n_2}^{n_1+n_2-1} \cdot C_{n_3}^{n_2+n_3-1}$ different possible kinematic analogs given n_1 elements of the first class, n_2 elements of the second class and n_3 trivial ele-

ments of the third class.

In general, the total number of different ray paths belonging to the group of kinematic analogs $(n_1,n_2,\ldots,n_J)$, $J \geq 2$ is given by

$$N_k(n_1,n_2,\ldots,n_J) = \prod_{j=1}^{J-1} C_{n_{j+1}}^{n_j+n_{j+1}-1} \tag{4}$$

1.4 Dynamic Analogs

As defined earlier, dynamic analogs are groups of rays that arrive at the receiver at identical times and with identical amplitudes. As was shown in Figure 1, a particular group of dynamic analogs exists as a subset of a specific group of kinematic analogs. Therefore the kinematic code, $(n_1,n_2,\ldots,n_J)$ must be identical for all elements of any particular group of dynamic analogs. In addition to sharing the same kinematic code, each element must be reflected m_j times, $j = 1,\ldots,J$, from the j^{th} interface. This ensures equality of amplitude.

For a complete description of a particular group of dynamic analogs (2J - 1) integers will be used. The first J integers will consist of the kinematic code which along with the velocity structure of the layered media determines the time of arrival at the surface. The kinematic code is followed by J-1 integers $(m_1,m_2,\ldots,m_{J-1})$ which represent the number of reflections from the j^{th} interface. The com-

plete code as shown in (5) allows the amplitude of the phase to be determined.

$$(n_1, n_2, \ldots, n_J; m_1, m_2, \ldots, m_{J-1}) \tag{5}$$

It will be noticed in (5) that the number of reflections, m_J, from the J^{th} or last interface, has not been specified. The reason is that $m_J = n_J$. To understand this, note that no normal element can exist in the J^{th} layer. The elements therefore, must be trivial. It can also be stated that for continuity, a trivial element must constitute a reflection. Since there are n_J half-segments in the J^{th} layer, there are n_J coupled segments which form n_J trivial elements and n_J reflections. Hence $n_J = m_J$.

The values which each m_j can take on is dependent on the values of n_j and n_{j+1}, $j = 1, J - 1$. To investigate the nature of this dependence recall that each reflection in layer j demands a trivial element of the j^{th} class. Recall also that at least one element of the j^{th} class must be normal. Hence for n_j elements of the j^{th} class, at most $n_j - 1$ can be trivial and there can be at most $n_j - 1$ reflections from the j^{th} interface. That is

$$m_j \leq n_j - 1 \tag{6}$$

Now consider two cases:

Case 1: $n_{j+1} \geq n_j$

If all n_j elements of the j^{th} class are normal then the number of reflections m_j from the j^{th} interface must equal zero. However it is only necessary that one element of the j^{th} class be normal so that for this case

$$m_j \geq 0. \tag{7}$$

Case 2: $n_{j+1} < n_j$

In this case not all n_j elements of the j^{th} class can be normal. The greatest allowable number of normal elements of the j^{th} class is n_{j+1}. Hence there must be at least $(n_j - n_{j+1})$ trivial elements of the j^{th} class. For this case then

$$m_j \geq (n_j - n_{j+1}). \tag{8}$$

Collecting (6), (7) and (8) yields

$$\mathrm{MAX}(0, n_j - n_{j+1}) \leq m_j \leq n_j - 1. \tag{9}$$

Using (9) it is possible to generate the symbols for all dynamic analogs given the kinematic code.

The number of dynamic analogs may be defined as the number of ray paths which satisfy the conditions inherent in the ray code, $(n_1, n_2, \ldots, n_J;\ m_1, m_2, \ldots, m_{J-1})$. In order to calculate this number consider first only the J^{th} and (J-1)st layer. The n_J trivial elements of the J^{th} class must be distributed among the n_{J-1} elements of the (J-1)st class leaving m_{J-1} of them trivial. The solution is given by the product of the number of different ways of choosing m_{J-1} trivial elements from the n_{J-1} elements of the (J-1)st layer, and the number of ways of distributing the m_J trivial elements of the J^{th} class among exactly $(n_{J-1} - m_{J-1})$ normal elements of the (J-1)st class. The problem is equivalent to the distribution of m_J balls into $(n_{J-1} - m_{J-1})$ pockets, none of which may be empty. The result which is derived in Appendix B is given by

$$\frac{n_{J-1}!}{m_{J-1}!(n_{J-1} - m_{J-1})!} \cdot \frac{(n_J - 1)!}{(n_J - n_{J-1} + m_{J-1})!(n_{j-1} - m_{J-1} - 1)!}$$

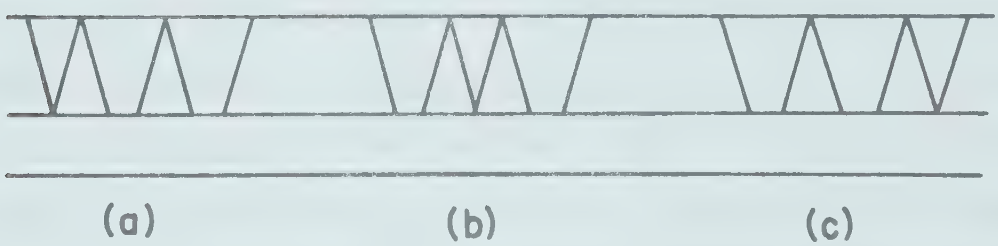
$$= C^{n_{J-1}}_{m_{J-1}} \cdot C^{n_J - 1}_{n_{J-1} - m_{J-1} - 1} \tag{10}$$

If the same reasoning is applied to the remaining layers the expression for the total number of dynamic analogs with the ray code $(n_1, n_2, \ldots, n_J;\ m_1, m_2, \ldots, m_{J-1})$ is

$$\prod_{j=1}^{J-1} C_{m_j}^{n_j} \cdot C_{n_j-m_j-1}^{n_{j+1}-1} \qquad (11)$$

A simple two-layer example with ray code (3,3;1) may serve to illustrate the procedure. From the ray code it is observed that there are three elements of the first class, one of which is trivial and three trivial elements of the second class. There are three different ways of choosing the single trivial element from the total of three elements of the first class. These are shown in Figure 5.

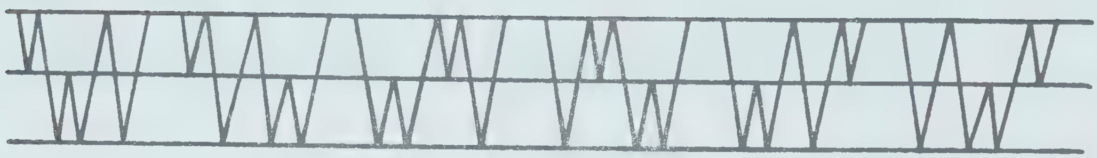
Figure 5. The Three Ways of Selecting One Trivial Element from Three Elements of the First Class.



The three trivial elements of the second class must now be distributed among the normal elements of the first class. For each of the three cases in Figure 5, there are two possible ways of doing this for a total of $C_1^3 \cdot C_1^2 = 3 \cdot 2 = 6$

dynamic analogs. These are shown in Figure 6.

Figure 6. The Dynamic Analogs with Code (3,3;1)



The advantages of the classification and coding scheme used by Hron should now be obvious when dealing with plane layers. Instead of calculating the amplitude contribution and time of arrival of each of the phases shown in Figure 6 separately, the amplitude and arrival time are calculated only once. The amplitude value when multiplied by the number of dynamic analogs as determined from (11) yields the total dynamic effect of the group. In terms of the number of computations, the savings grow as the number of segments increase to accommodate a greater number of layers. In such cases the number of dynamic analogs for a particular ray code can easily reach several hundreds.

1.5 Hron's Selection Criteria and Algorithm for the Generation of Ray Codes

Before designing an algorithm to generate ray codes

as described in the previous sections it is necessary to select a family of rays from the infinite number of possible choices. Hron chooses that family of ray paths such that a maximum of HMS half-segments are allowed in any one phase where HMS is a specified input parameter. Table 1 shows typical output using this criteria for HMS = LAY = 4. LAY represents the number of layers in the system.

Table 1. Typical Output for HMS = LAY = 4

Group Number	Ray Code
1	(1)
2	(2)
3	(1,1;0)
4	(3)
5	(2,1;1)
6	(1,2;0)
7	(1,1,1;0,0)
8	(4)
9	(3,1;2)
10	(2,2;0)
11	(2,2;1)
12	(1,3;0)
13	(2,1,1;1,0)
14	(1,2,1;0,1)
15	(1,1,2;0,0)
16	(1,1,1,1;0,0)

If HSG represents half the number of segments in a particular group of ray paths, then HSG can take on values from 1 to HMS. These HSG half-segments can be distributed among the top LM layers where LM = MIN(HSG,LAY). The restriction on LM is to ensure continuity. If LBT is the number of

the deepest layer penetrated by the ray with HSG half segments, there are as many different groups of kinematic analogs as there are possible distributions of HSG coupled segments among LBT layers leaving none of them empty. According to Appendix B this number is given by

$$M(LBT,HSG) = \frac{(HSG - 1)!}{(LBT - 1)!(HSG - LBT)!} = C^{HSG-1}_{LBT-1} \tag{12}$$

In accordance with the selection criteria, LBT is allowed to vary between 1 and LAY and HSG varies between 1 and HMS. Therefore the total number of different groups of kinematic analogs is given by

$$M_t(LAY,HMS) = \sum_{HSG=1}^{HMS} \sum_{LBT=1}^{LM} M(LBT,HSG)$$

$$= \sum_{HSG=1}^{HMS} \sum_{LBT=1}^{LM} C^{HSG-1}_{LBT-1} \tag{13}$$

In (13) consider the term $C^{HMS-1}_{LM-1} = C^{HMS-1}_{MIN(HMS,LAY)-1}$ where LBT = LM and HSG = HMS. Since the condition $HMS - 1 \geq MIN(HMS,LAY) - 1$ must hold in order for the term to have a non-zero value it may be rewritten as C^{LMX-1}_{LMX-1} where LMX = MIN(HMS,LAY). Expanding (13) now yields

$$\sum_{HSG=1}^{HMS} \sum_{LBT=1}^{LM} \sum_{LBT-1}^{HSG-1} = C_0^0 + C_0^1 + C_1^1 + C_0^2 + \ldots$$

$$+ C_{LMX-3}^{LMX-1} + C_{LMX-2}^{LMX-1} + C_{LMX-1}^{LMX-1} \qquad (14)$$

Recalling that Pascal's Triangle tabulates the values of C_r^n, it may be noted that if n = HSG - 1 and r = LMX - 1 then (14) represents the summation of up to LMX values in each of the first HSG rows of Pascal's Triangle. For example in Table 2 the sum of the values within the boxed area represent the total number of kinematic analogs in six layers with the maximum number of half-segments, HSG, being equal to eight.

In order to generate the codes for the ray paths governed by the selection criteria, a matrix N(I,L) is set up, each row containing a different kinematic code. The column dimension must be set equal to MIN(HMS,LAY) but although the total number of kinematic analogs is given by (13) it is not necessary that the row dimension assume this value. Instead, a reduced row dimension may be substituted. Consider Table 3 which was obtained from the last eight rows of Table 1. The last column in Table 3 with the heading M(LBT,4) is seen to correspond to the fourth row of Pascal's Triangle. The three rows in Group II were all obtained from row 8 by subtracting integer values 1, 2 and 3 from the first column and adding them to the second column.

Table 2. Pascal's Triangle - C_r^n

	LMX →	1	2	3	4	5	6	7	8	9	10	11
	r(LMX-1) →	0	1	2	3	4	5	6	7	8	9	10
HMS	n(HMS-1) ↓											
1	0	1										
2	1	1	1									
3	2	1	2	1								
4	3	1	3	3	1							
5	4	1	4	6	4	1						
6	5	1	5	10	10	5	1					
7	6	1	6	15	20	15	6	1				
8	7	1	7	21	35	35	21	7	1			
9	8	1	8	28	56	70	56	28	8	1		
10	9	1	9	36	84	126	126	84	36	9	1	
11	10	1	10	45	120	210	252	210	120	45	10	1

Similarly row 12 was created from row 10 and row 13 from row 11 by subtracting 1 from the second column and adding it to the third. Row 14 was created from row 11 by subtracting the value 2 from column two and adding it to column three. Finally row 15 was obtained from row 14 by subtracting 1 from column three and adding it to column four.

Table 3. Based on Hron (1972). Demonstration of an Algorithm for the Creation of Symbols for Kinematic Analogs.

Row Number	Row Vector		LBT	M(LBT,4)
8	4	- I	1	1
9	3,1			
10	2,2	- II	2	3
11	1,3			
12	2,1,1			
13	1,2,1	- III	3	3
14	1,1,2			
15	1,1,1,1	- IV	4	1

Since group (i) may be generated from group (i-1), i = II, IV, in the example shown, the maximum number of rows necessary in the matrix N(I,L) is equal to twice the maximum value of M(LBT,4). In this case the value equals 2 * 3 = 6. In general the reduced row dimension DIM is given by

$$DIM = 2 * MAX\left(\frac{(HMS - 1)!}{(L - 1)!(HMS-L)!}\right) \tag{15}$$

where

$$1 \leq L \leq MIN(HMS, LAY).$$

A comparison between the actual number of kinematic analogs and the reduced row dimension DIM is given in Table 4.

Table 4. Comparison of the Total Number of Kinematic Analogs $M_t(LAY,HMS)$ and the Reduced Row Dimension DIM for HMS = LAY.

LAY	2	3	4	5	6	7	8	9	10	11	12
M_t(LAY,HMS)	3	7	15	31	63	127	255	511	1023	2047	4095
DIM	2	4	6	12	20	40	70	140	252	504	924

CHAPTER II

THE PRACTICAL APPLICATION OF THE RAY METHOD OF HRON (1972)

2.1 The Matrix N(I,L)

In the last chapter it was demonstrated that by using the concept of dynamic analogs a saving in time could be realized when calculating the amplitude contributions of the various phases. Despite this advantage there is a serious limitation in the algorithm for the generation of the kinematic code as presented in 1.5. The limitation lies in the amount of additional space necessary for the matrix N(I,L) as the number of layers are increased. Table 5 shows

Table 5. Total Space Required for N(I,L)

L(LAY)	I(DIM)	I·L
2	2	4
4	6	24
6	20	120
8	70	560
10	252	2,520
12	924	11,088
14	3,432	48,048
16	12,870	205,920
18	48,620	875,160
20	184,756	3,695,120
22	705,432	15,519,504
24	2,704,156	64,899,744

the total amount of space (I·L) needed for the matrix N(I,L) when HMS = LAY, LAY = 2(2)24. From the table it can be seen that the amount of space required for the matrix N(I,L) grows very quickly. It is clear that if a model consisting of many layers is to be used, a different approach would be useful to generate the kinematic code more efficiently.

2.2 Partitions, Compositions and a New Algorithm for the Generation of the Kinematic Code

A partition is defined as a collection of integers with a given sum without regard to order; the corresponding ordered collections are called compositions. The integers which form a partition or composition are called its parts and the number which is the sum of these parts is known as the partitioned or composed number. By convention repeated parts are abbreviated by the use of exponents. For example 222 is written as 2^3. Table 6 shows the unrestricted partitions and compositions of the numbers 1 to 4.

Table 6. The Unrestricted Partitions and Compositions of the Numbers 1 to 4.

Number	Partitions	Compositions
1	1	1
2	2,11	2,11
3	3,21,1^3	3,21,12,1^3
4	4,31,2^2,21^2,1^4	4,31,13,2^2,211,121,112,1^4

Restrictions can be imposed on partitions and compositions as to the kind (even or odd), size, number of repetitions, or number of parts. Table 7 shows all partitions of n for n = 1(1)8 grouped by the number of parts.

Table 7. Partitions of n by Number of Parts from Riordan (1958).

	Number of Parts							
n	1	2	3	4	5	6	7	8
1	1							
2	2	1^2						
3	3	21	1^3					
4	4	31	21^2	1^4				
		2^2						
5	5	41	31^2	21^3	1^5			
		32	2^21					
6	6	51	41^2	31^3	21^4	1^6		
		42	321	2^21^2				
		3^2	2^3					
7	7	61	51^2	41^3	31^4	21^5	1^7	
		52	421	321^2	2^21^3			
		43	3^21	2^31				
			32^2					
8	8	71	61^2	51^3	41^4	31^5	21^6	1^8
		62	521	421^2	321^3	2^21^4		
		53	431	3^21^2	2^31^2			
		4^2	42^2	32^21				
			3^22	2^4				

Recall from the previous chapter that the selection criteria of Hron (1972) entails the distribution of up to HMS segments among a maximum of MIN(HSG,LAY) layers. For each value of HSG the resulting distribution yields a collection of kinematic codes. The integers in each code from this collection must sum to HSG. Hence the collection of kinematic codes represents a composition of HSG with the restriction that the number of parts of the composition,

$m \leq$ MIN(HSG,LAY). Equivalently the collection of codes may be viewed as the result of first partitioning HSG followed by generating the unique permutations of each partition. For example the composition of the integer 4 in Table 6 is identical to the kinematic code (row vectors) in Table 3 of the previous chapter. Thus the basis for a new algorithm to generate the kinematic code is established. A general procedure for the generation of synthetic seismograms using this approach is shown in Figure 7.

2.3 The Partitioning Algorithm

In the computer program written to generate synthetic seismograms by this method, the partitioning of the integers 1 to HMS was carried out using an algorithm presented by Lehmer (1964). The flowchart for this subroutine is shown in Figure 8. This procedure generates those partitions whose number of parts, m, satisfy $p \leq m \leq q$. In order to generate all partitions of a given integer, p is set equal to 1 and q is assigned the value n where n is the integer being partitioned. In this manner all partitions of n are generated beginning with n and ending with 1,1,...,1. The vector "a" of length q stores the partition currently being generated and a variable T in conjunction with m ensures that this partition will be unique. The generation of the current partition or kinematic code does not require the manipulation of previously generated codes. This results in great savings of storage space as will be seen.

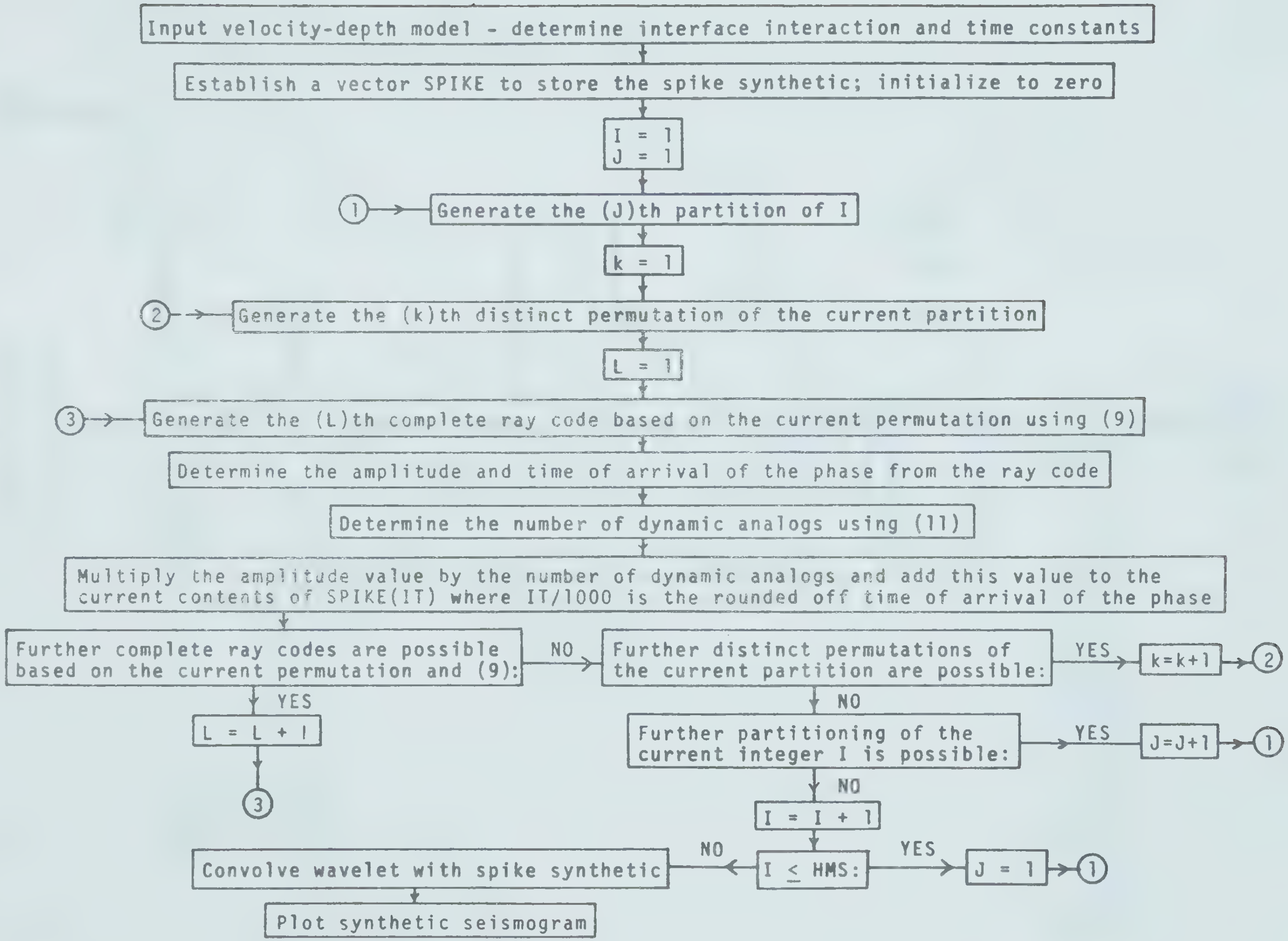
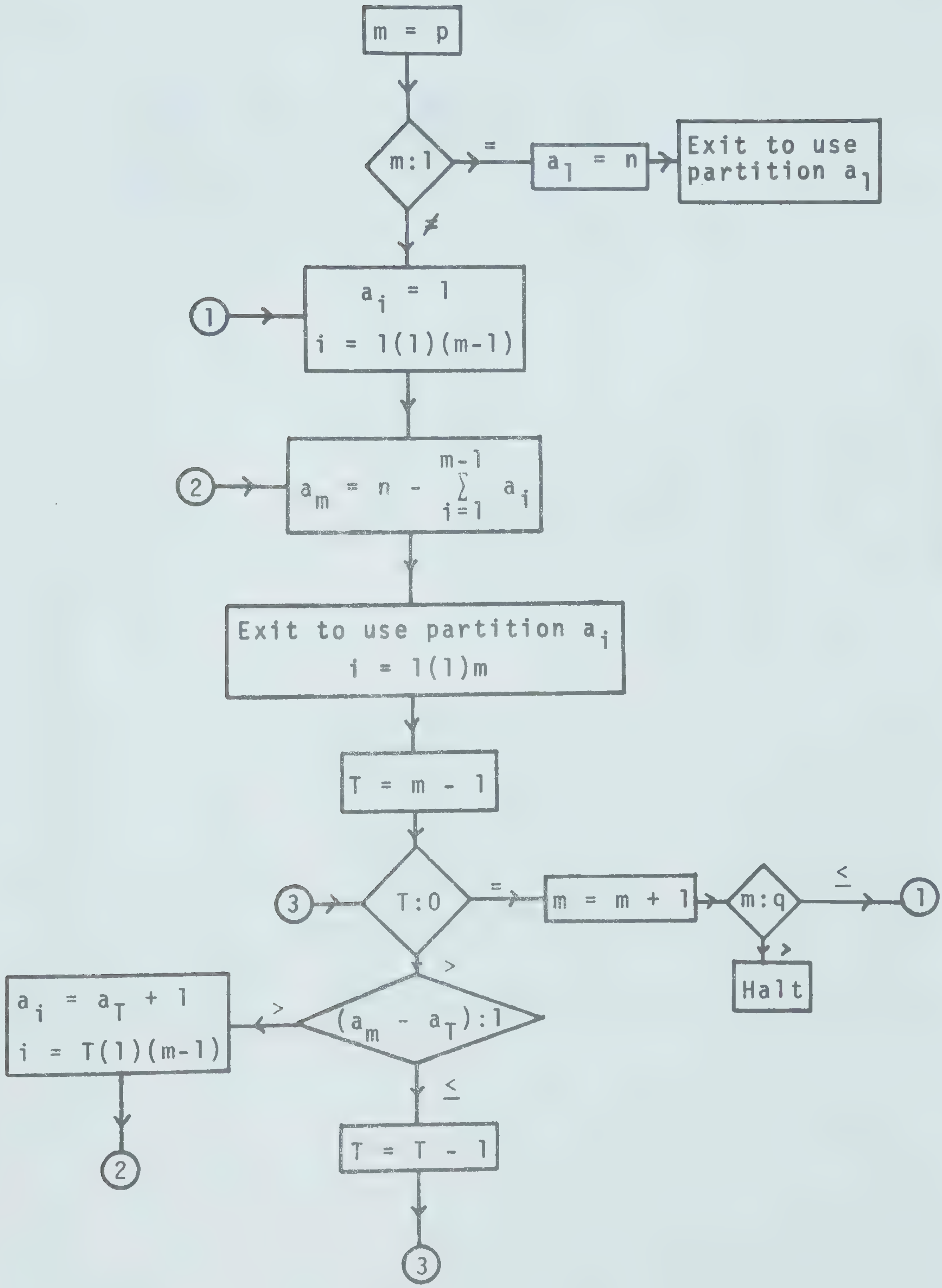


Figure 7. Procedure for the Generation of Synthetic Seismograms.

Figure 8. The Partitioning Algorithm.



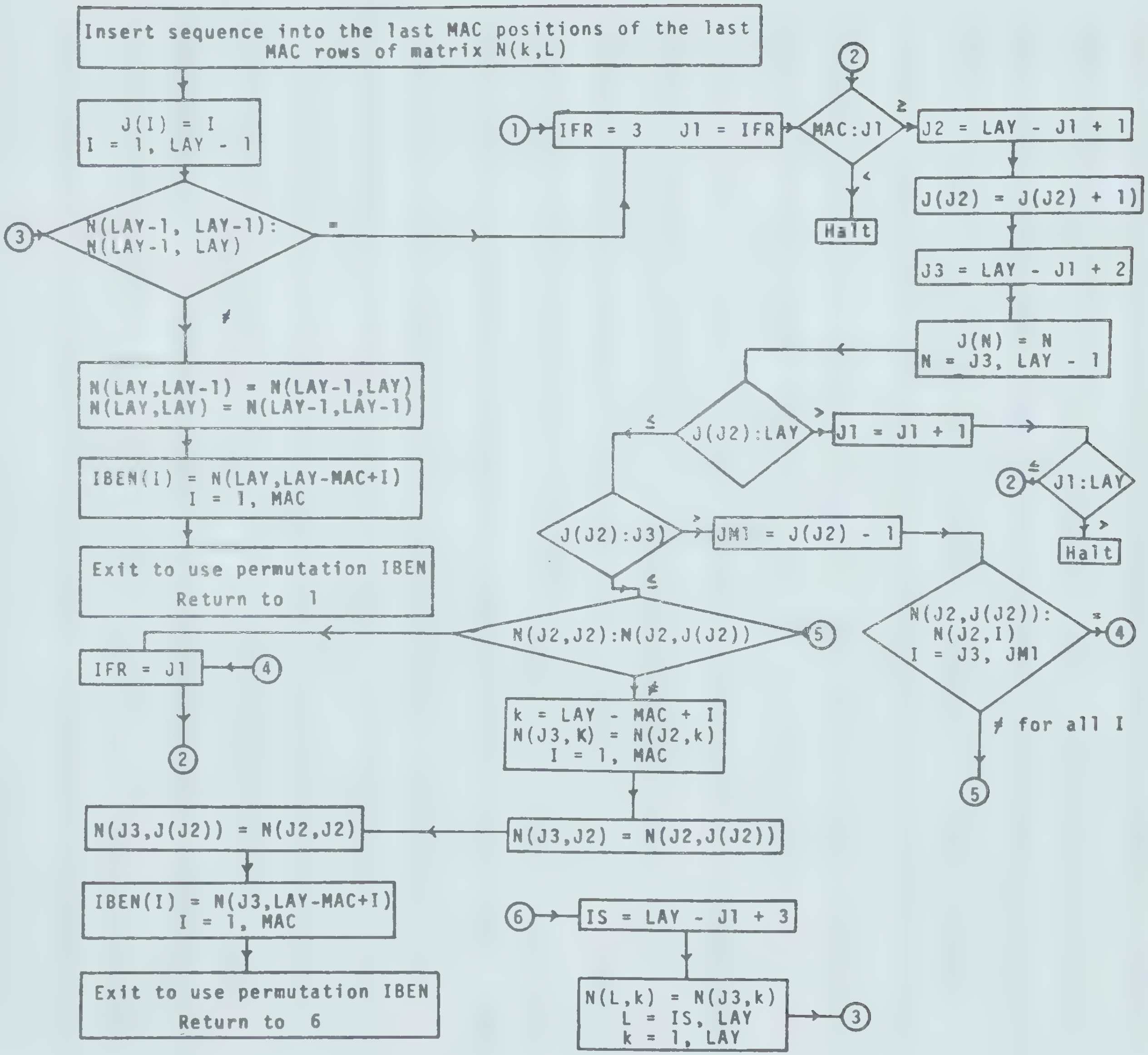


Figure 9. The Permutation Algorithm.

2.4 The Permutation Subroutine

As seen in Figure 7, after each partition is obtained the distinct permutations of the partition are generated. Each of these permutations including the original partition represent a particular kinematic code. There are several algorithms available for the efficient generation of permutations, e.g., Paige *et al.* (1960), Johnson (1963), Wells (1961). They are all based on the fact that there are n! permutations of n distinct marks and also n! (n-1)-digit numbers of the form: $a_1 1! + a_2 2! + \ldots + a_{n-1}(n-1)!$, $0 \leq a_i \leq i$, $i = 1(1)(n-1)$. The permutations are generated by setting up a one-to-one correspondence between the n! permutations and the n! signatures $(a_1, a_2, \ldots, a_{n-1})$. Despite being efficient both in terms of processing time and storage costs, a totally different approach was necessary. The reason is that these methods are designed to produce all the permutations of n *distinct* marks. If the marks are not all distinct as is the case generally when dealing with partitions, repeated codes will be generated when using these methods. In order to prevent this from happening and still retain this approach, large amounts of storage space would appear to be necessary. Hence a different method of producing the distinct permutations of a given partition was developed and the flowchart is presented in Figure 9.

The permutation subroutine makes use of a matrix N(K,L), the column and row dimensions both being equal to the number of layers (LAY) in the model render considera-

tion. Before beginning, the current partition is inserted into the last MAC positions of each row in the matrix where MAC represents the number of parts in the partition. The algorithm is constructed in such a way that all distinct permutations involving only the last k integers of the partition are generated before permutations involving only the last k + 1 integers are attempted, k = 2(1)(MAC-1). This is done through the use of a vector J(I) whose value indicates the column position of the last component in the partition that the integer in column I of row I has attempted to exchange places with. The value of J(I) is always greater than I and the interchange of integers is always between positions I and J(I) in row I of the matrix N(k,L). In order to ensure that the permutations will be distinct two criteria are used. The first which should be obvious, is that no two integers that have the same value will be interchanged. The second criterion is that if J(I) $\geq$ I + 1 and N(I,J(I)) = N(I,k), k = (I+1), (J(I)-1) then no interchange is made. A simple example may serve to illustrate the general procedure for the generation of permutations using this method.

Consider the partition 2133 which has been written into the last four positions of the last four rows of the matrix N(k,L). Suppose also that LAY = 10 so that the dimensions of the matrix are 10 x 10. The initial values of J(I) and that portion of the matrix which will be used to generate the permutations are shown in Table 8.

Table 8. Initialization of Matrix N(k,L) and Vector J(I).

Column → / Row ↓	7	8	9	10
7	2	1	3	3
8	2	1	3	3
9	2	1	3	3
10	2	1	3	3

J(7) = 7

J(8) = 8

J(9) = 9

Original Partition

2133 (1)

The value of J(9) is increased by one to 10 so that the integers are to be interchanged in positions N(9,9) and N(9,J(9)), that is, in positions N(9,9) and N(9,10). They are equal therefore no exchange is made. The value J(8) is now increased by one to 9 and J(9) is returned to its original value of 9. Integers in positions N(8,8) and N(8,J(8)) are to be interchanged. This operation is performed and the resulting permutation is written into rows 9 and 10 as shown in Table 9. Row 8 remains unaltered. J(9) is now increased

Table 9. Generation of Sequence (2)

Column → / Row ↓	7	8	9	10
7	2	1	3	3
8	2	1	3	3
9	2	3	1	3
10	2	3	1	3

J(7) = 7

J(8) = 9

J(9) = 9

Permutation Generated

2313 (2)

by one to 10 and the integers in positions N(9,9) and N(9,J(9)) are to be interchanged. The operation is performed and the resulting permutation is written into row 10 as shown in Table 10. Row 9 remains unchanged. J(8) is now increased by 1 to 10 and J(9) is returned to its' original value. Integers in positions N(8,8) and N(8,J(8)) are to be interchanged. No interchange is made due to the second

Table 10. Generation of Sequence (3)

Column → Row ↓	7	8	9	10		Permutation Generated
					J(7) = 7	
					J(8) = 9	2331 (3)
7	2	1	3	3		
8	2	1	3	3	J(9) = 10	
9	2	3	1	3		
10	2	3	3	1		

criterion given previously for the generation of distinct permutations. If the interchange were made, the result would be identical to sequence (3). Since J(8) cannot be increased beyond 10, it is returned to its' initial value of 8 and J(7) is increased by one to 8. Integers in positions N(7,7) and N(7,J(7)) are to be interchanged. Once this is done all distinct permutations involving integers in colums 8, 9 and 10 are generated as previously shown. The value of J(7) is then increased by one and the procedure is repeated. This continues until all permutations involving J(7) = 10 have been generated. Following this example,

the operation of the algorithm should be clear.

2.5 Cost Comparison Between Hron's Algorithm and the Permuted Partition Approach

The total amount of vector space necessary for the generation of the kinematic code using the permuted partition approach is given by $(LAY)^2 + 3(LAY)$. These values are tabulated and compared with the space needed for Hron's matrix N(I,L) in Table 11. As the number of layers is increased it can be seen that the saving in space becomes

Table 11. Comparison of Vector Space Requirements for the Generation of the Kinematic Code.

LAY	N(I,L)	$(LAY)^2 + 3(LAY)$	
2	4	10	
4	24	28	
6	120	54	
8	560	88	
10	2,520	130	
12	11,088	180	HMS = LAY
14	48,048	238	
16	205,920	304	
18	875,160	378	
20	3,695,120	460	
22	15,519,504	550	
24	64,899,744	648	

enormous using the permuted partition approach. The reduced amount of space should result in a more economical means of generating synthetic seismograms. In order to test this, two computer programs were written to generate synthetic

seismograms at vertical incidence. They were identical in all respects except in their method of generating the kinematic code. One program used the algorithm of Hron (1972), the other made use of the permuted partition approach. For this test the maximum allowable number of half-setments (HMS) was set equal to LAY. The resulting cost breakdown is shown in Table 12.

For models of twelve layers or less the space necessary for the generation of the kinematic code in both programs is considerably less than the necessary peripheral space; that is the space required for the model parameters, spike synthetic, wavelet storage, and the plotting routines. Therefore in the case of the nine and twelve layer models the difference in storage costs between the two programs is rather small. However, in the case of the fourteen and sixteen layer models, the required space for the matrix N(I,L) has grown very rapidly beyond the peripheral storage requirements. In addition, the difference between the values $(LAY)^2 + 3(LAY)$ and (I)(L) for LAY = 16 is much larger than the corresponding difference for LAY = 9. The result is a reduction in total computing costs by approximately one-half when using the permuted partition approach for sixteen layers and HMS = LAY.

2.6 Restricted Partitioning

Despite the reduction in cost obtained by using the method of permuted partitions the generation of the synthetic

seismogram for sixteen layers with HMS = LAY was still relatively expensive. It would be useful therefore to examine the effect of further restricting the number of phases that contribute to the final result.

It should be noted that the ray selection criteria of Hron (1972) used in the cost comparison studies are themselves a restriction, as is setting the value of HMS equal to the number of layers (LAY). The result is an incomplete or partial ray expansion, where only those rays that are thought to contribute large amplitudes are included. M. Hron (1973) studied the problem of ray series convergence in some detail and was unable to find any theoretical criteria by which the ray series could be terminated with assurance that the remainder would be negligible. The result of this is that any partial ray expansion must be the result of intuition as to which group of rays contribute the maximum amplitude. With this in mind further restrictions on the partial ray expansion of Hron (1972) are examined.

In order to reduce the number of phases, restrictions on the number of parts allowed in a given partition were imposed in four different levels of severity. These are shown in Figure 10. The partitions to the right of the stepped line in each of Figure 10(a), (b), (c) and (d) are used in the computation while the partitions to the left are discarded. This approach tends to eliminate phases with a large number of reflections by disallowing the distri-

Figure 10. Restricted Partitions

Number of Parts

n	1	2	3	4	5	6	7	8
1	1							
2	2	1^2						
3	3	21	1^3					
4	4	31	21^2	1^4				
		2^2						
5	5	41	31^2	21^3	1^5			
		32	2^21					
6	6	51	41^2	31^3	21^4	1^6		
		42	321	2^21^2				
		3^2	2^3					
7	7	61	51^2	41^3	31^4	21^5	1^7	
		52	421	321^2	2^21^3			
		43	3^21	2^31				
			32^2					
8	8	71	61^2	51^3	41^4	31^5	21^6	1^8
		62	521	421^2	321^3	2^21^4		
		53	431	3^21^2	2^31^2			
		4^2	42^2	32^21				
			3^22	2^4				

(a) Severity Level 1

Number of Parts

n	1	2	3	4	5	6	7	8
1	1							
2	2	1^2						
3	3	21	1^3					
4	4	31	21^2	1^4				
		2^2						
5	5	41	31^2	21^3	1^5			
		32	2^21					
6	6	51	41^2	31^3	21^4	1^6		
		42	321	2^21^2				
		3^2	2^3					
7	7	61	51^2	41^3	31^4	21^5	1^7	
		52	421	321^2	2^21^3			
		43	3^21	2^31				
			32^2					
8	8	71	61^2	51^3	41^4	31^5	21^6	1^8
		62	521	421^2	321^3	2^21^4		
		53	431	3^21^2	2^31^2			
		4^2	42^2	32^21				
			3^22	2^4				

(b) Severity Level 2

Number of Parts

n	1	2	3	4	5	6	7	8
1	1							
2	2	1^2						
3	3	21	1^3					
4	4	31	21^2	1^4				
		2^2						
5	5	41	31^2	21^3	1^5			
		32	2^21					
6	6	51	41^2	31^3	21^4	1^6		
		42	321	2^21^2				
		3^2	2^3					
7	7	61	51^2	41^3	31^4	21^5	1^7	
		52	421	321^2	2^21^3			
		43	3^21	2^31				
			32^2					
8	8	71	61^2	51^3	41^4	31^5	21^6	1^8
		62	521	421^2	321^3	2^21^4		
		53	431	3^21^2	2^31^2			
		4^2	42^2	32^21				
			3^22	2^4				

(c) Severity Level 3

Number of Parts

n	1	2	3	4	5	6	7	8
1	1							
2	2	1^2						
3	3	21	1^3					
4	4	31	21^2	1^4				
		2^2						
5	5	41	31^2	21^3	1^5			
		32	2^21					
6	6	51	41^2	31^3	21^4	1^6		
		42	321	2^21^2				
		3^2	2^3					
7	7	61	51^2	41^3	31^4	21^5	1^7	
		52	421	321^2	2^21^3			
		43	3^21	2^31				
			32^2					
8	8	71	61^2	51^3	41^4	31^5	21^6	1^8
		62	521	421^2	321^3	2^21^4		
		53	431	3^21^2	2^31^2			
		4^2	42^2	32^21				
			3^22	2^4				

(d) Severity Level 4

bution of a large number of half-segments into a small number of layers. The restriction on the number of reflections could of course be applied directly when generating the dynamic portion of the ray code, however it would then be necessary to produce the complete set of partitions. In the method described the suspect partitions are eliminated directly by simply not generating them resulting in a greater saving of CPU time.

The restricted partitioning was first applied to a sixteen layer model, (Figure 11), which approximates the geologic structure near Edmonton. The true structure of the area was obtained from a digitized sonic log of well CS Big Hay Lake located at 13-30-48-22W4. The log provided data for the interval between 670 and 6,030 feet. The problem of how to best approximate a complex velocity structure such as that obtained from the digitized sonic log by a model with a small number of layers is not dealt with here.

In Figure 12 the spike synthetics obtained using the sixteen layer model are shown for the unrestricted and four different levels of restricted partitions. The total number of different phases represented in each of these spike sequences along with the costs involved in producing the final synthetic seismograms are presented in Table 13. There is a noticeable change in the character of the spike sequences in progressing from the unrestricted case to Severity Level 4. In particular there is a large reduction in the number of phases arriving at times greater than 1.105

Figure 11. Sixteen Layer Model.

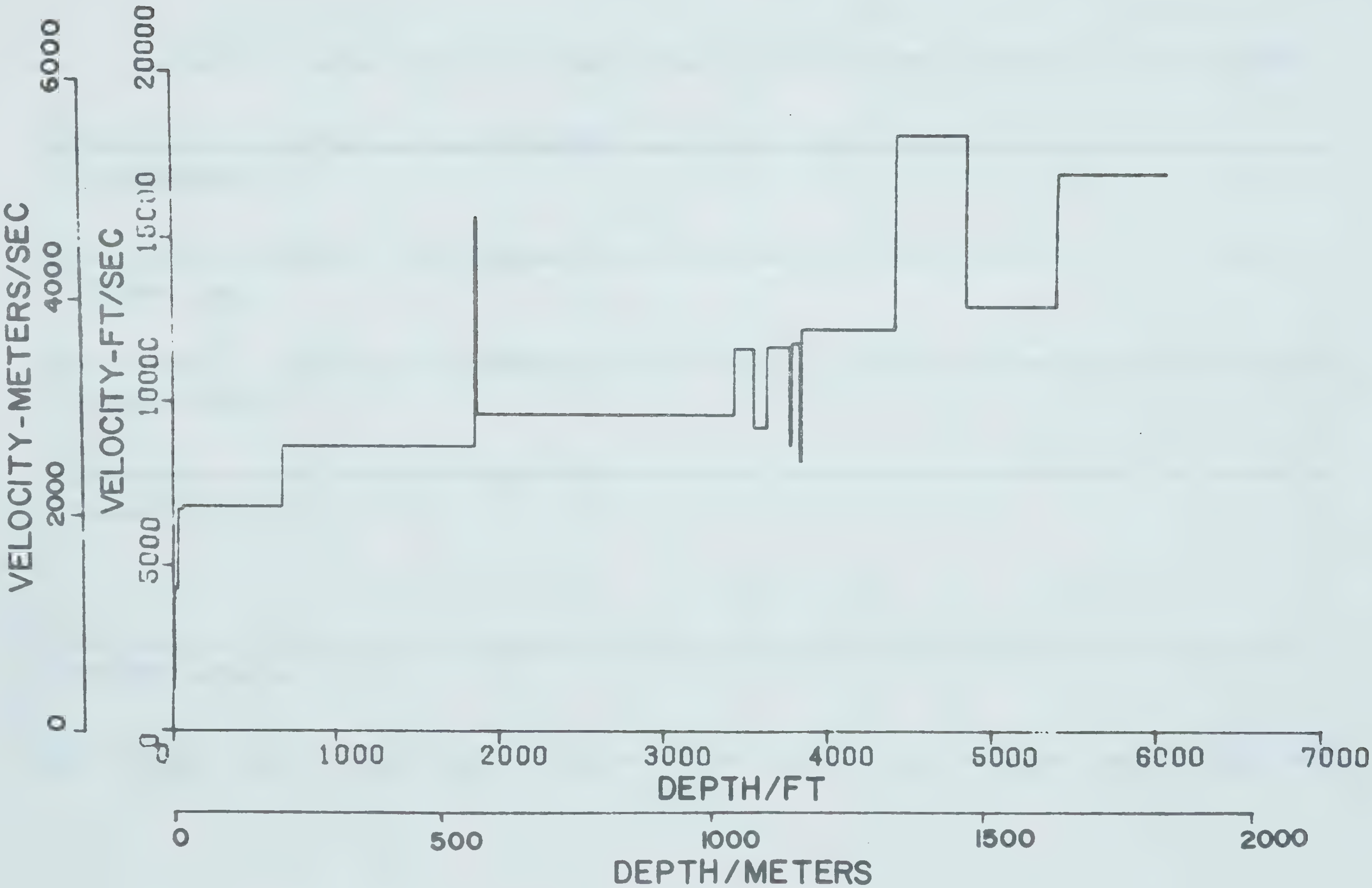


Figure 12. Spike Synthetics Generated Using Different Levels of Restriction.

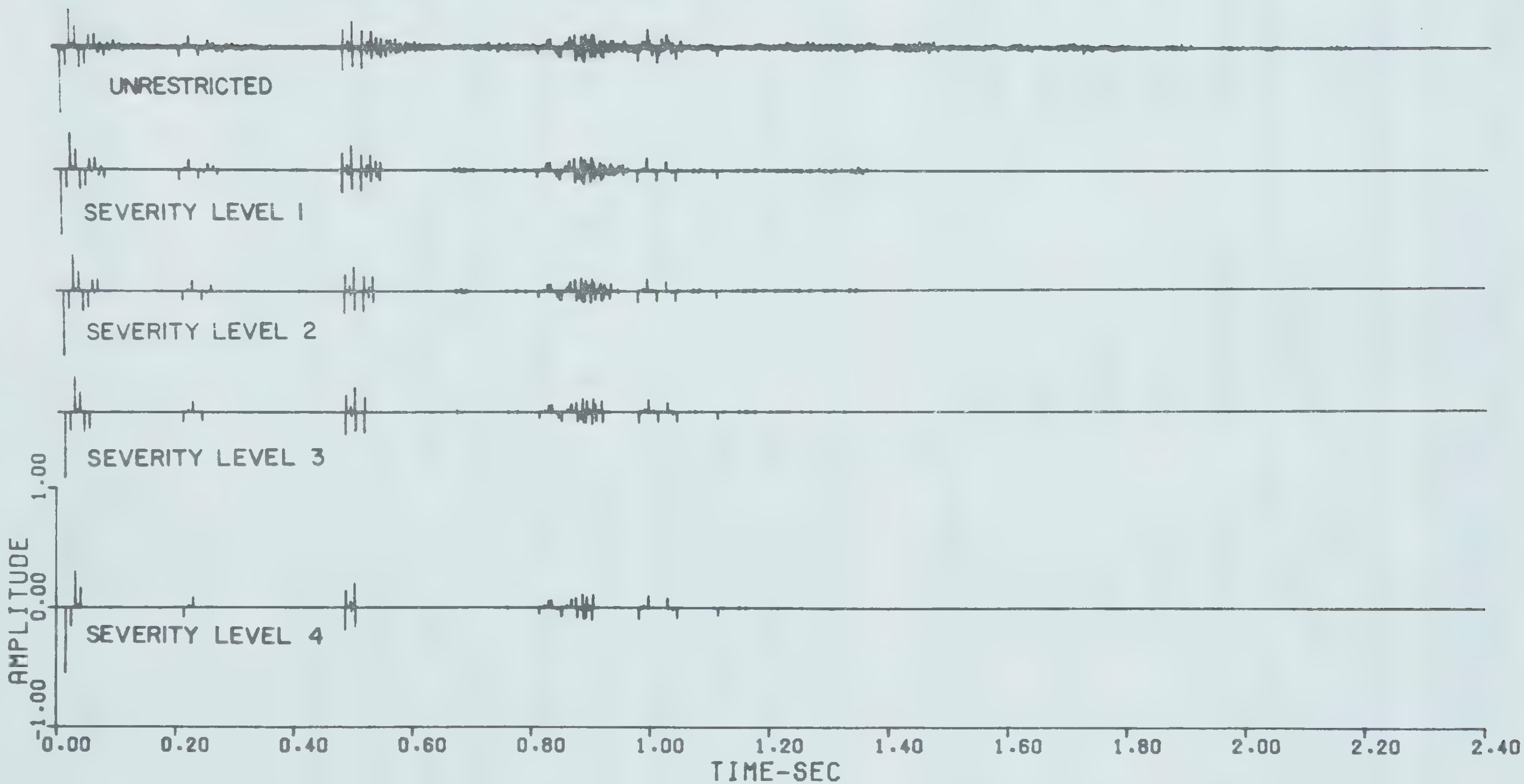


Table 13. Cost Comparison of Four Levels of Restriction in the Generation of the 16-Layer Synthetic Seismogram.

	Total Number of Phases	CPU Time (sec.)	CPU Storage (pg-min.)
Unrestricted	48,760,342	254.8	292.0
Severity Level 1	87,668	17.8	19.4
Severity Level 2	16,343	9.6	10.0
Severity Level 3	2,335	7.0	7.2
Severity Level 4	241	5.9	5.9

seconds which marks the end of the velocity structure measured in two-way travel time. Hence the restricted partitions have been successful in terms of reducing the number of late arriving phases with a corresponding drastic reduction in overall computing costs. However, there has also been a reduction in the number of phases arriving at times of less than 1.105 seconds and the question remains as to how severe this reduction has been. In order to obtain a quantitative answer to this question a relative percentage difference was calculated for each .2 second time interval. That is, for each interval:

$$\text{Average Relative Percentage Difference} = \frac{1}{n} \sum_i \frac{|U_i - R_i|}{|U_{MAX}|} \times 100\% \qquad (16)$$

where U_i is the amplitude value of the spike in the unrestricted sequence at time i/1000 and R_i is the amplitude value of the spike in a restricted sequence at the identical time; U_{MAX} is the amplitude value of the largest spike from the unrestricted sequence in the time interval, and n is the number of non-zero values of U_i within the interval. If the value U_i is found to be equal to zero the operation $|U_i - R_i|/|U_{MAX}|$ is not performed. The reason for using the expression $|U_i - R_i|/|U_{MAX}|$ in (16) rather than the usual percentage difference given by $|U_i - R_i|/|U_i|$ is that the latter does not yield a good measure of the dissimilarity between the two final synthetic seismograms. This is because the absence of very small spikes in a region characterized by spikes of much larger amplitude, has only minor effects on the final trace, yet the values $|U_i - R_i|/|U_i|$ could be very large giving a false indication of poor agreement between the two seismograms. The measure in (16) was applied to all four severity levels in Figure 12 and the results are shown in Table 14. The bracketed quantities give the maximum relative percentage differences within each time interval. Some of these have high values which indicate that there are spikes of significant amplitude which have been excluded by the restriction process. The values in the last column of Table 14 give a quantitative measure of the spike sequence degeneration as seen in Figure 12. To investigate the dependence of the averages over 1.105 seconds on the length of the time interval chosen,

Table 14. Average Relative Percentage Difference for the 16-Layer Model

	Average Relative Percentage Differences						
Time Interval (sec.)	0-.2	.2-.4	.4-.6	.6-.8	.8-1.0	1.0-1.105	Average Over 1.105 sec.
Severity Level 1	.7 (15)	2.0 (36)	2.0 (36)	11.9 (100)	4.8 (100)	9.7 (62)	5.1
Severity Level 2	1.0 (23)	2.5 (41)	2.8 (41)	12.4 (100)	6.1 (100)	10.0 (62)	5.8
Severity Level 3	1.5 (33)	3.0 (60)	3.8 (61)	13.4 (100)	7.7 (100)	10.2 (62)	6.7
Severity Level 4	2.2 (55)	3.6 (82)	4.7 (83)	14.0 (100)	9.5 (100)	11.3 (119)	7.6
Number of Non-Zero Samples	154	166	177	200	200	105	

Table 14 was recalculated using a time interval of .1 seconds. The changes noted were not significant.

The final synthetic seismograms were produced by convolving a wavelet with the spike sequences in Figure 12. These are shown along with the sonigram in Figure 13. Although the general features are present throughout, the character changes in going from the unrestricted wave train to Severity Level 4.

2.7 Synthetic Seismograms with Geometric Spreading

The spike sequences in Figure 12 were generated without taking into account the factor introduced into the amplitude as a result of geometric spreading of the wavefront. One advantage of the ray tracing method is that inclusion of this factor is a relatively simple matter. Synthetic seismograms based on a 14-layer model were generated both with and without geometric spreading in order to observe any changes in character. The spike sequences were generated letting HMS = LAY and the wavelet used in generating the final traces was identical to the one shown in Figure 12. The spike synthetics and convolved traces that resulted from unrestricted partitioning are shown in Figure 14 along with the sonigram.

It is clear that if the spreading correction is applied to the spike sequence in Figure 14(a), the changes in amplitude of the phases will not occur in a uniform or predictable manner when viewed in time. This is confirmed

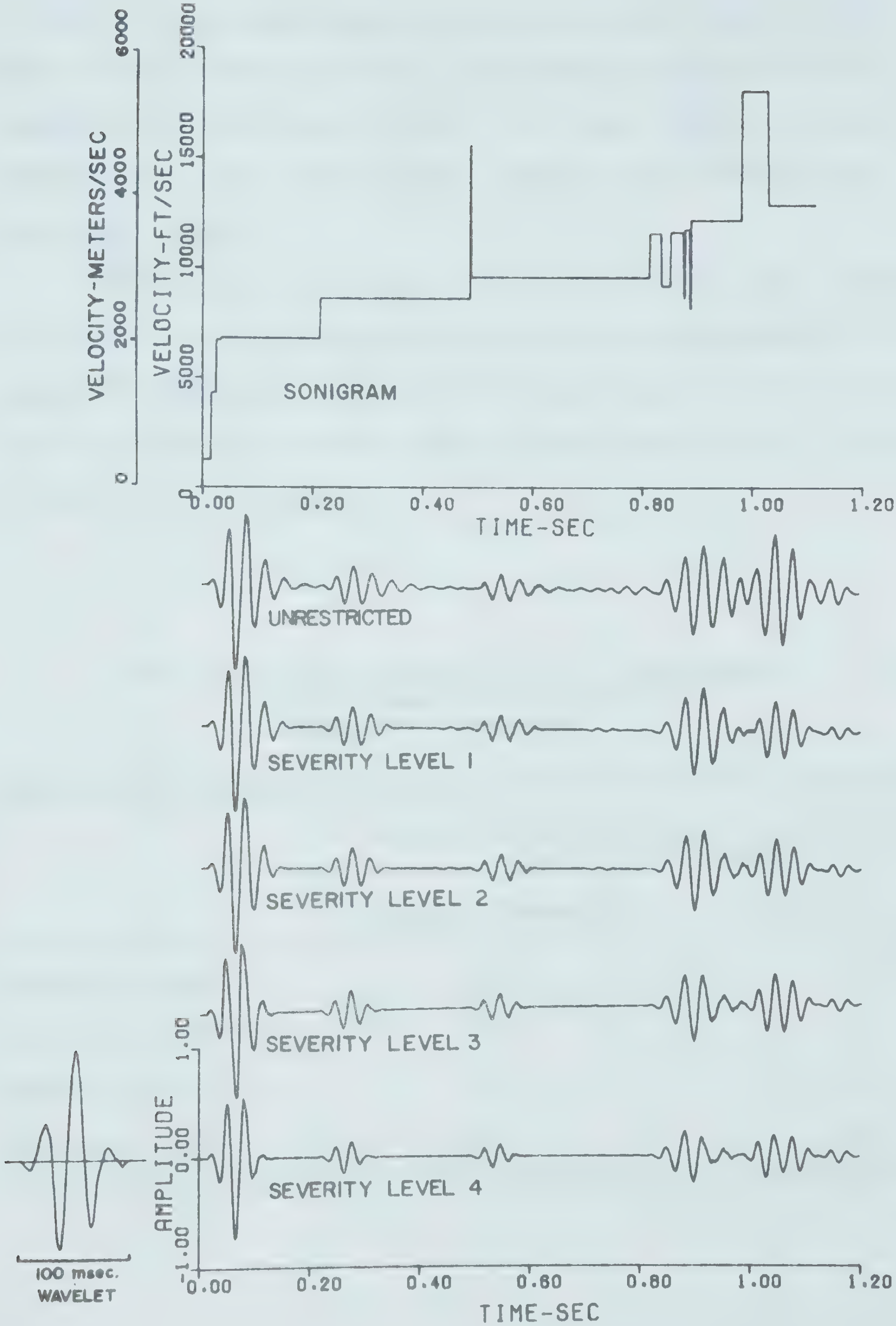


Figure 13. The Sonigram and Synthetic Seismograms for the 16-Layer Model.

by noting the large change in relative amplitudes of the group of arrivals at .2 and .5 seconds in Figure 14(b) as compared with the same arrivals in Figure 14(a). In the remainder of the spike sequence however, the differences are rather subtle.

The measure of dissimilarity used in the last section was applied to the results of restricted partitioning with geometric spreading using the 14-layer model. The process was then repeated without the spreading correction. Table 15 displays the results.

Table 15. The Average Relative Percentage Difference With and Without Geometric Spreading.

	Average Relative Percentage Differences Over 1.105 Seconds	
	Without Spreading	With Spreading
Severity Level 1	3.9	3.8
Severity Level 2	4.7	4.7
Severity Level 3	5.7	5.6
Severity Level 4	6.6	6.5

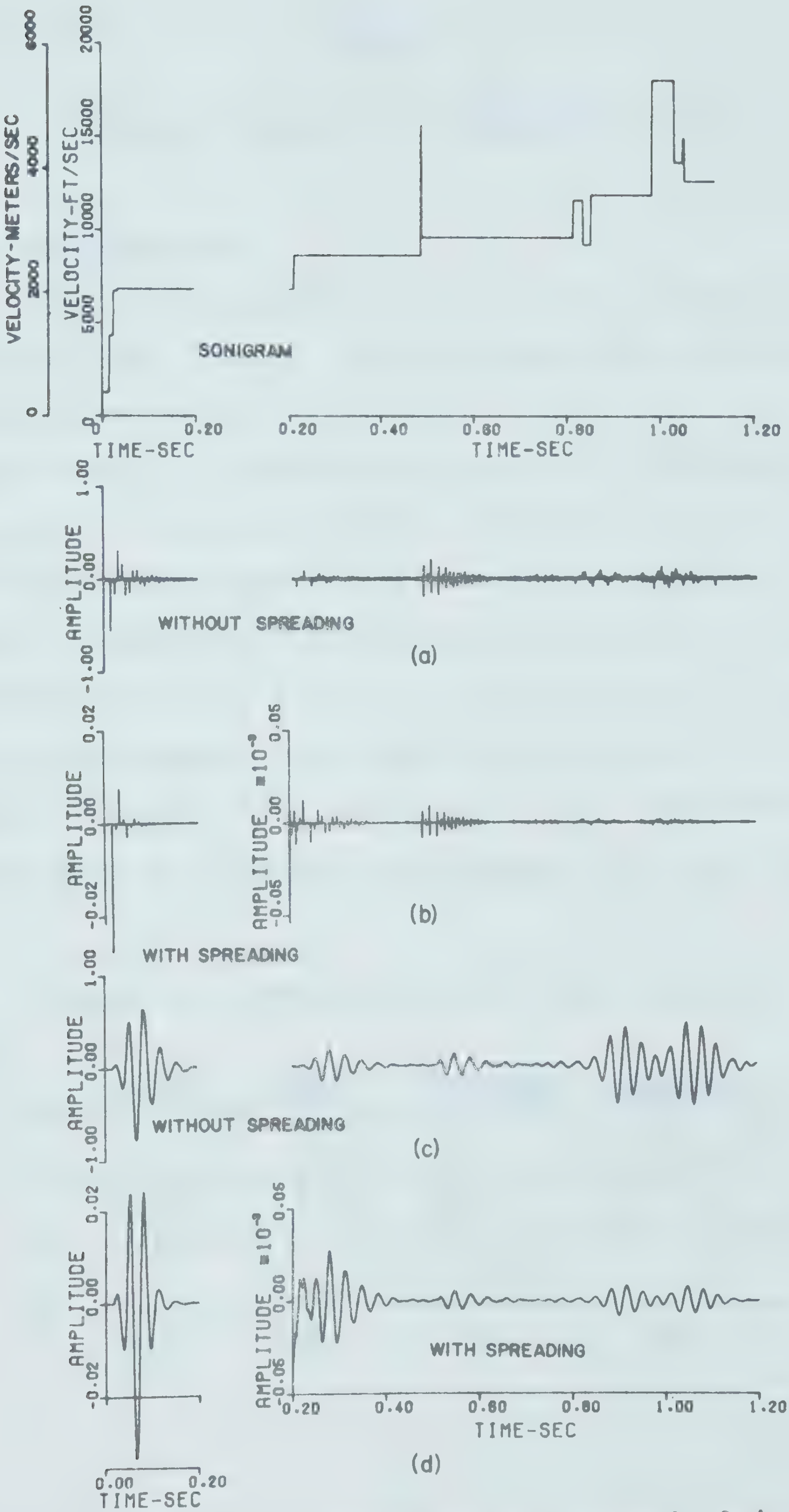


Figure 14. A Comparison of Synthetic Seismograms With and Without Geometric Spreading.

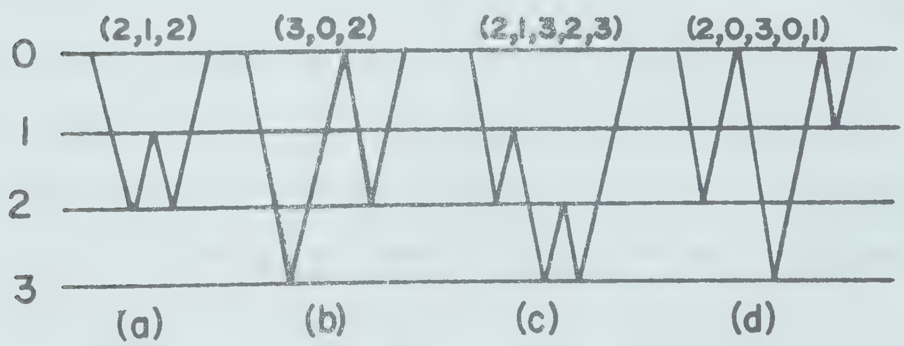
CHAPTER III

THE RAY METHOD OF N^{th} ORDER MULTIPLES

3.1 Basic Concepts

The last two chapters were centered around the ray method of Hron (1972). In this chapter the classification scheme commonly used in industry is described. This classification is based on the number of reflections and the groups of ray paths which result are called n^{th} order multiples. The order of the multiple is defined as the number of downward reflections suffered by the ray in travelling from the source to receiver; that is, the number of times an upgoing ray segment is transformed into a down going ray segment by a reflection from an interface. Figure 15 may help to illustrate the concept. The ray paths shown

Figure 15. First and Second Order Multiples.



in Figure 15(a) and (b) are both first order multiples while those in (c) and (d) are second order multiples.

Although the method described in Chapter I and the n^{th} order multiple approach are based on entirely different criteria, the two sets containing the phases for each scheme using a j-layer velocity model, will in general have a fairly large intersection. The degree to which this hold true depends on the specification of the value HMS and the highest order of multiples to be used. It is entirely possible for one set to exist as a subset of the other. For example by specifying HMS = 2 * LAY, the first order multiples are automatically included using the scheme described in the first chapter. Letting HMS = 4 * LAY includes first and second order multiples.

3.2 The Coding Scheme and Method of Generation

The code for a particular ray path using the approach of n^{th} order multiples consists of $2n + 1$ integers where n represents the order of the multiple. Each integer m(i) of the code represents the interface from which the i^{th} reflection occurs. Some typical ray paths and their codes have already been seen in Figure 15. For first order multiples (Figure 15(a) and (b)) the range of values for m(i), i = 1, 3 is given by:

$$1 \le m(1) \le LAY$$

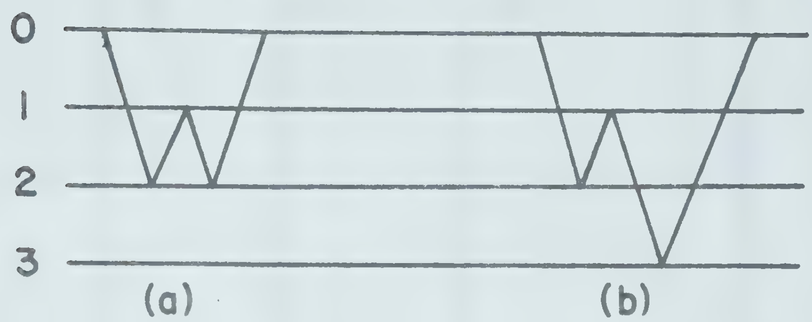
$$0 \le m(2) \le m(1) - 1 \qquad (17)$$

$$m(2) + 1 \le m(3) \le LAY$$

The range of values for the k^{th} order multiples may be specified in a similar manner.

In order to generate the codes, the sequence of inequalities in (17) were written into nested "do-loops". Using this method, vector space is needed only for storing the code currently being generated. As in the scheme described in Chapters I and II, each code completely specifies the kinematic and dynamic properties of a single phase travelling from the source to receiver. However in the n^{th} order multiple approach at vertical incidence using the code generation scheme described, it is not necessary to determine the amplitude of each of these phases separately. In the computer program written to generate synthetic seismograms using this approach, the amplitude of the phase currently being generated is obtained from the amplitude of the previous phase. To understand how this is possible consider the two phases drawn in Figure 16. According to the scheme used, the phase shown in Figure 16(b) is generated immediately following the phase shown in (a). To obtain the amplitude of the phase in (b) it is necessary to divide the amplitude of the phase in (a) by the reflection coefficient at interface 2, multiply by the downward

Figure 16. Two Sequential First Order Multiples.

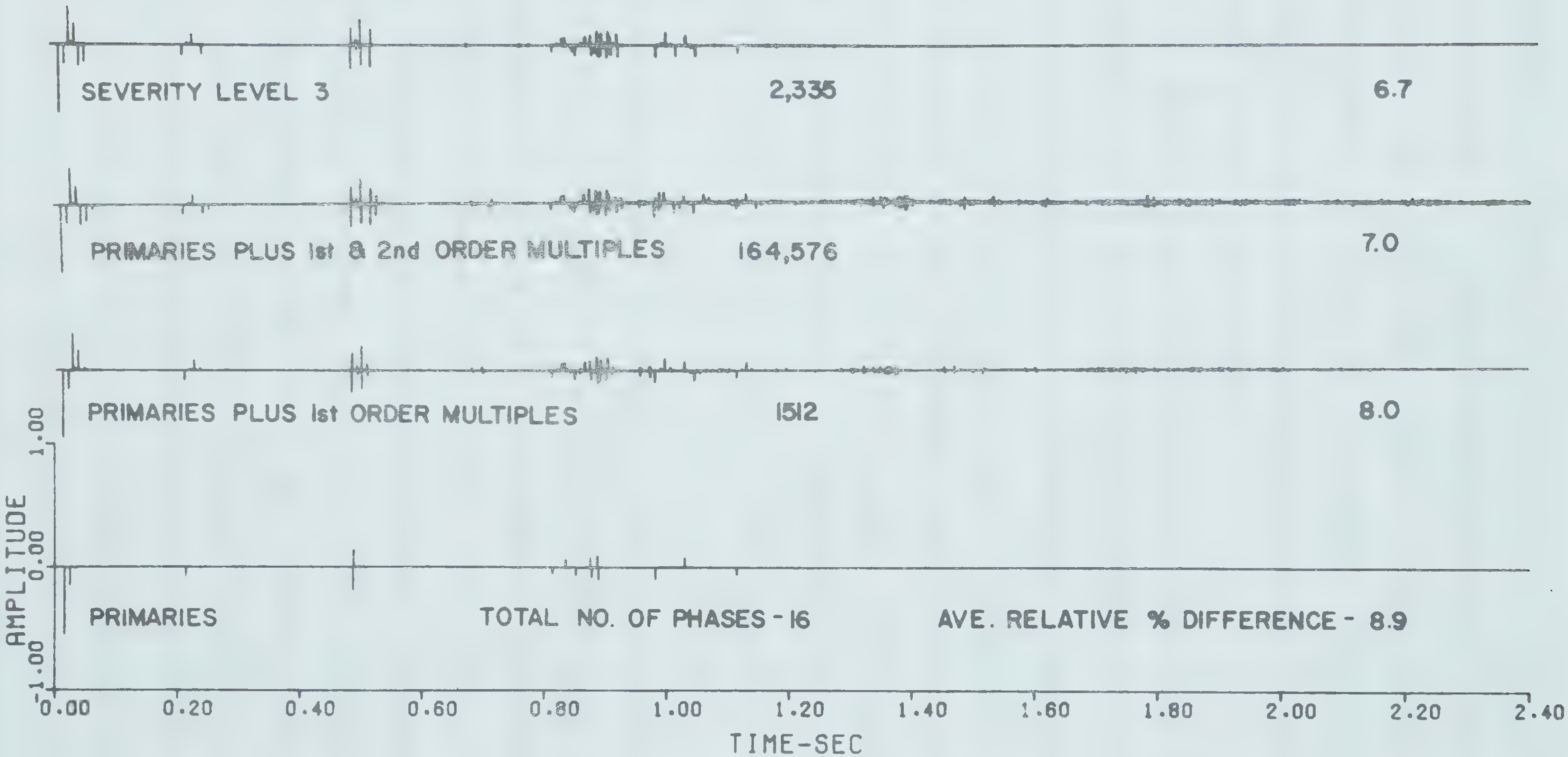


and upward transmission coefficients at interface 2 and finally multiply by the reflection coefficient at interface 3. Since calculating the amplitude in (b) would require eight operations if performed independently of (a), a saving of four operations has been realized. The savings increase as the higher order multiples are included. Unfortunately this saving may be overshadowed by the fact that the concept of dynamic equivalence has not been applied to this scheme.

3.3 Comparison of the Method of N^{th} Order Multiples and the Ray Method of Hron (1972)

A comparison was made between the two different ray classification schemes using the 16-layer model shown in Figure 11. The spike sequences for primaries plus up to second order multiples are shown in Figure 17 together with the Severity Level 3 spike synthetic. The average percent-

Figure 17. Spike Sequences for Severity Level 3 and Primaries Plus up to Second Order Multiples



age differences indicated are relative to the Unrestricted sequence in Figure 12. These values for each .2 second interval up to the end of the velocity structure at 1.105 seconds, are shown in Table 16. Bracketed quantities in Table 16 give the maximum relative percentage differences encountered in the time intervals.

It is interesting to note that the number of phases in the primaries plus first and second order multiples spike sequence is almost two orders of magnitude larger than the number of phases in the Severity Level 3 sequence. Despite this, the Severity Level 3 sequence has a smaller relative percentage difference over 1.105 seconds. The reason is that most of the contributions of the primaries plus first and second order multiples occur at times greater than 1.105 seconds.

The wavelet shown in Figure 13 was convolved with the spike sequences in Figure 17 to yield the synthetic seismograms displayed in Figure 18. The costs of generating these synthetics is shown in Table 17. The effective cost per phase in generating the Severity Level 3 sequence is 3.0×10^{-3} seconds compared with 6.1×10^{-5} seconds per phase for the primaries plus first and second order multiple sequence. The large difference occurred despite the fact that the generation of the primaries plus first and second order multiples sequence was done without making use of dynamic equivalence. It might be speculated that if restrictions were imposed on the primaries plus first and second

Table 16. Average Relative Percentage Differences Using N^{th} Order Multiples.

	Average Relative Percentage Differences						
Time Interval (sec.)	0-.2	.2-.4	.4-.6	.6-.8	.8-1.0	1.0-1.105	Average Over 1.105 Sec.
Severity Level 3	1.5 (33)	3.0 (60)	3.8 (61)	13.4 (100)	7.7 (100)	10.2 (62)	6.7
Primaries Plus 1st and 2nd Order	1.6 (33)	3.0 (60)	3.8 (61)	13.2 (100)	8.2 (100)	12.3 (65)	7.0
Primaries Plus 1st Order	2.3 (55)	3.7 (82)	4.9 (83)	14.2 (100)	10.4 (100)	11.8 (119)	8.0
Primaries Only	3.3 (100)	4.3 (100)	5.9 (100)	14.2 (100)	12.3 (100)	12.4 (119)	8.9
Number of Non-Zero Samples	154	166	177	200	200	105	

Figure 18. Synthetic Seismograms for Severity Level 3 and Primaries Plus up to Second order Multiples

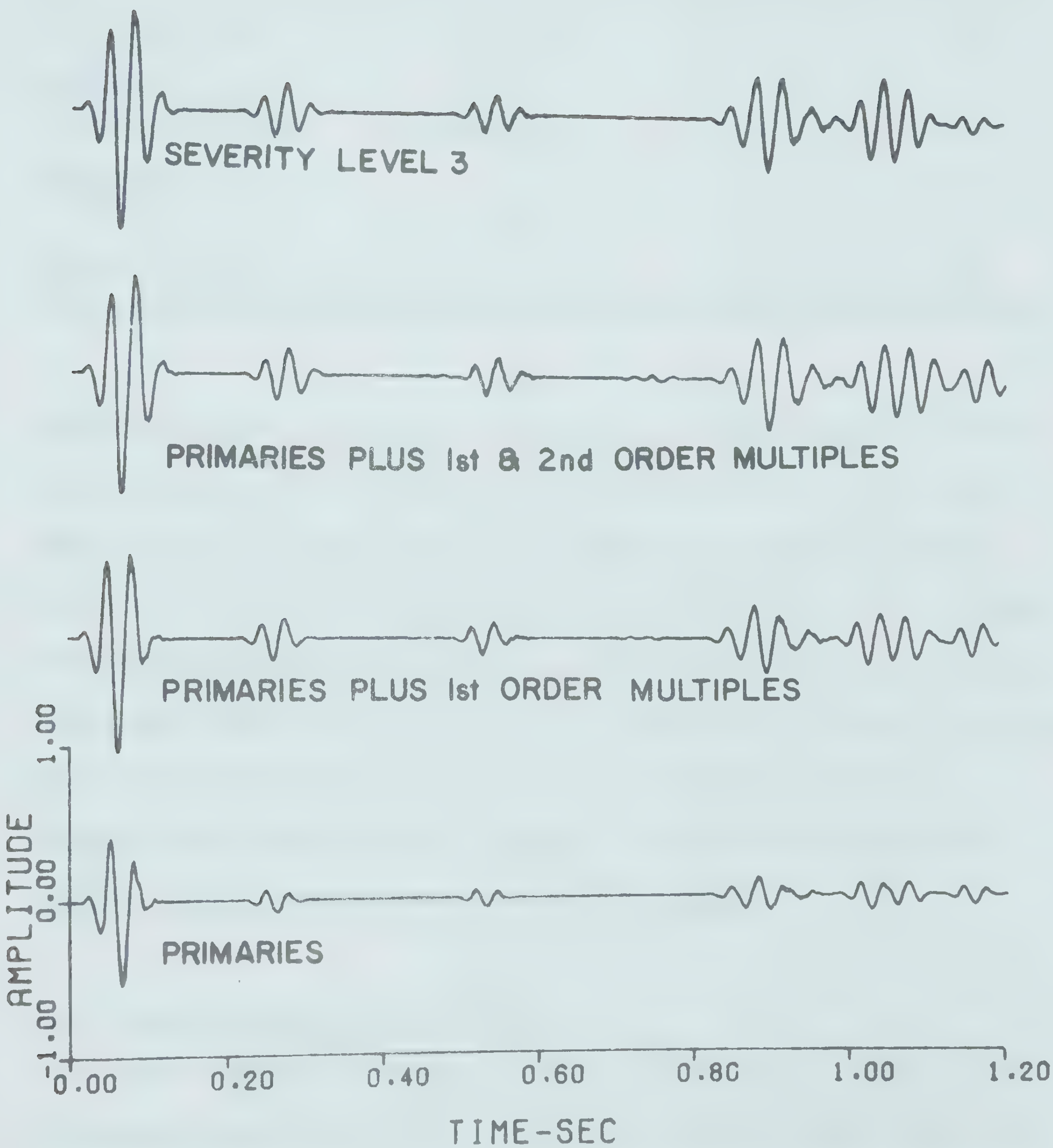


Table 17. Cost Comparison of Generating Primaries Plus Up to Second Order Multiples for the 16-Layer Model.

	Total Number Of Phases	CPU Time (sec.)	CPU Storage (pg-min.)
Severity Level 3	2,335	7.0	7.2
Primaries Plus 1st and 2nd Order	164,576	10.0	10.9
Primaries Plus 1st Order	1,512	5.7	6.0
Primaries Only	16	1.9	5.8

order multiples generating scheme in such a way that more of the phases were chosen to lie within the 1.105 second time window, then a restricted version of the n^{th} order multiple approach might be more economical than the scheme described in Chapters I and II. Unfortunately no investigations were made in this area. It may be of interest to note that the effective cost per phase in generating the Unrestricted sequence in Figure 12 using the concept of dynamic equivalence was 5.2×10^{-6} seconds.

3.4 Surface Order Multiples

N^{th} order "surface" multiples occur as a subset of n^{th} order multiples and are defined as those multiples in which $m(i) = 0$ for i even. For examples refer to Figure 15(b) and (d). Surface multiples have often been used as

an approximation to the complete set in designing multiple suppression algorithms, e.g., Watson (1965). To observe how well the approximation holds a comparison was performed between n^{th} order multiples and n^{th} order surface multiples up to the second order using the 16-layer model. Figure 19 shows the spike sequence and Figure 20 the final synthetic seismograms. The average percentage differences indicated in Figure 19 are relative to the Unrestricted spike sequence in Figure 12. The wavelet used to generate the synthetics was identical to the one shown in Figure 13. Agreement between the surface multiples and the complete set up to the second order is seen to be quite good in the case of the model used.

3.5 The Complete Wave Solution for the 16-Layer Model

To this point in discussing the various restrictions and schemes used in generating spike sequences, comparisons were generally made with the Unrestricted sequence in Figure 12. The reason for this is that this sequence represented the effect of the largest number of phases and was assumed to be the best approximation to the complete solution obtained by distributing up to an infinite number of half segments into a maximum of LMX layers. However if the Unrestricted sequence does not adequately represent the complete solution, then the comparisons made are of little significance. For this reason a spike sequence representing the complete wave solution was obtained and compared with

Figure 19. Comparison of Spike Sequences for N^{th} Order Multiples and N^{th} Order Surface Multiples

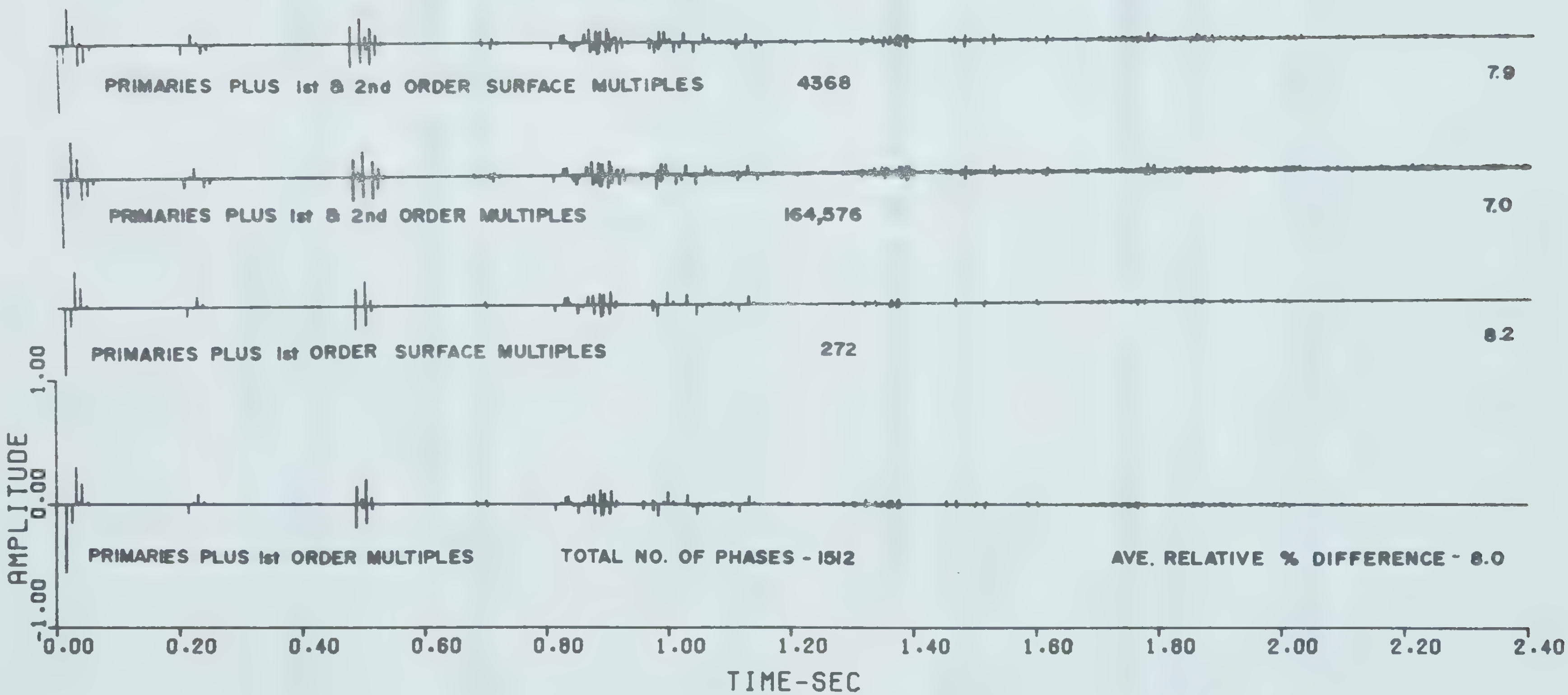
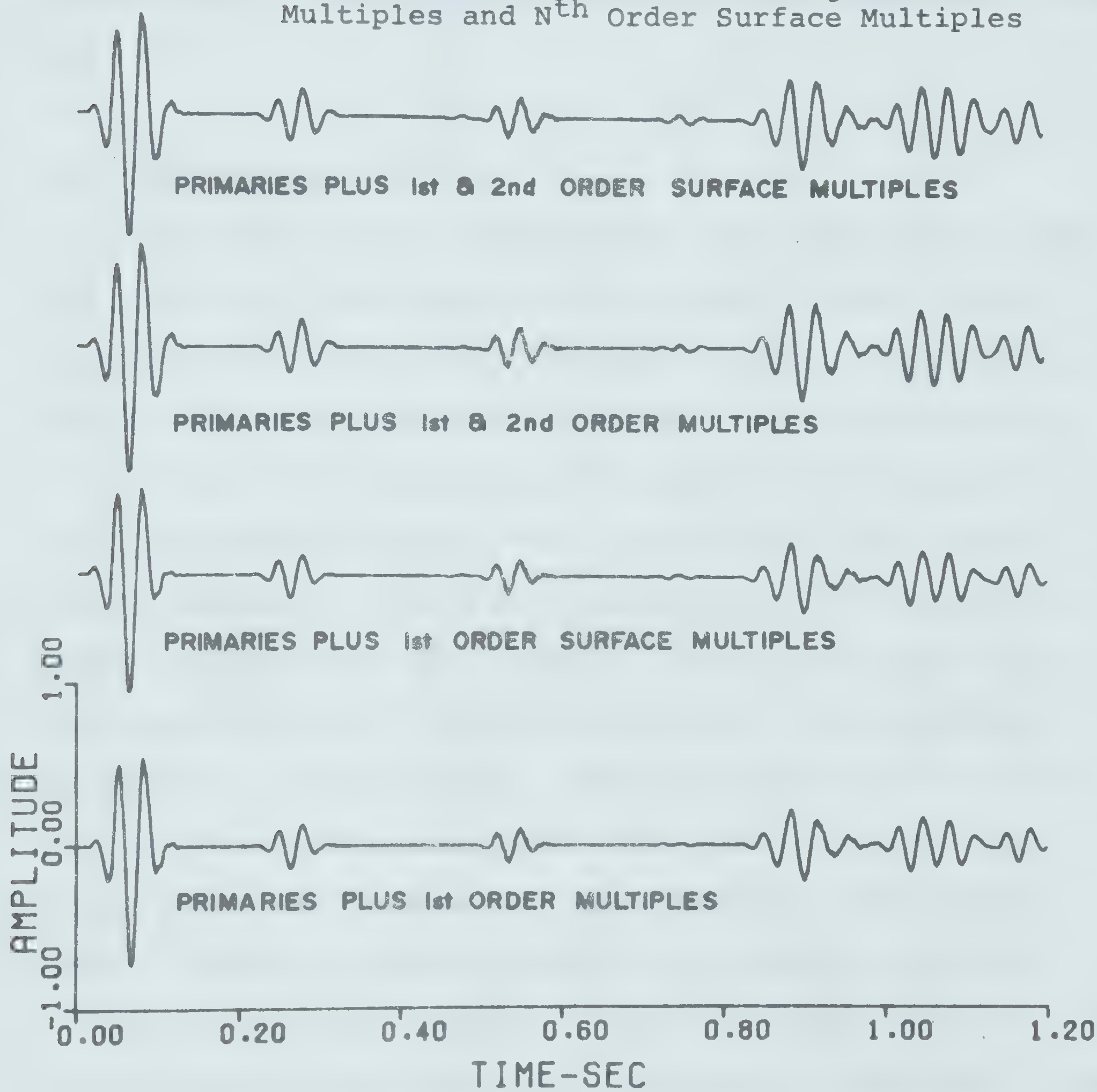


Figure 20. Comparison of Synthetic Seismograms for N^{th} Order Multiples and N^{th} Order Surface Multiples



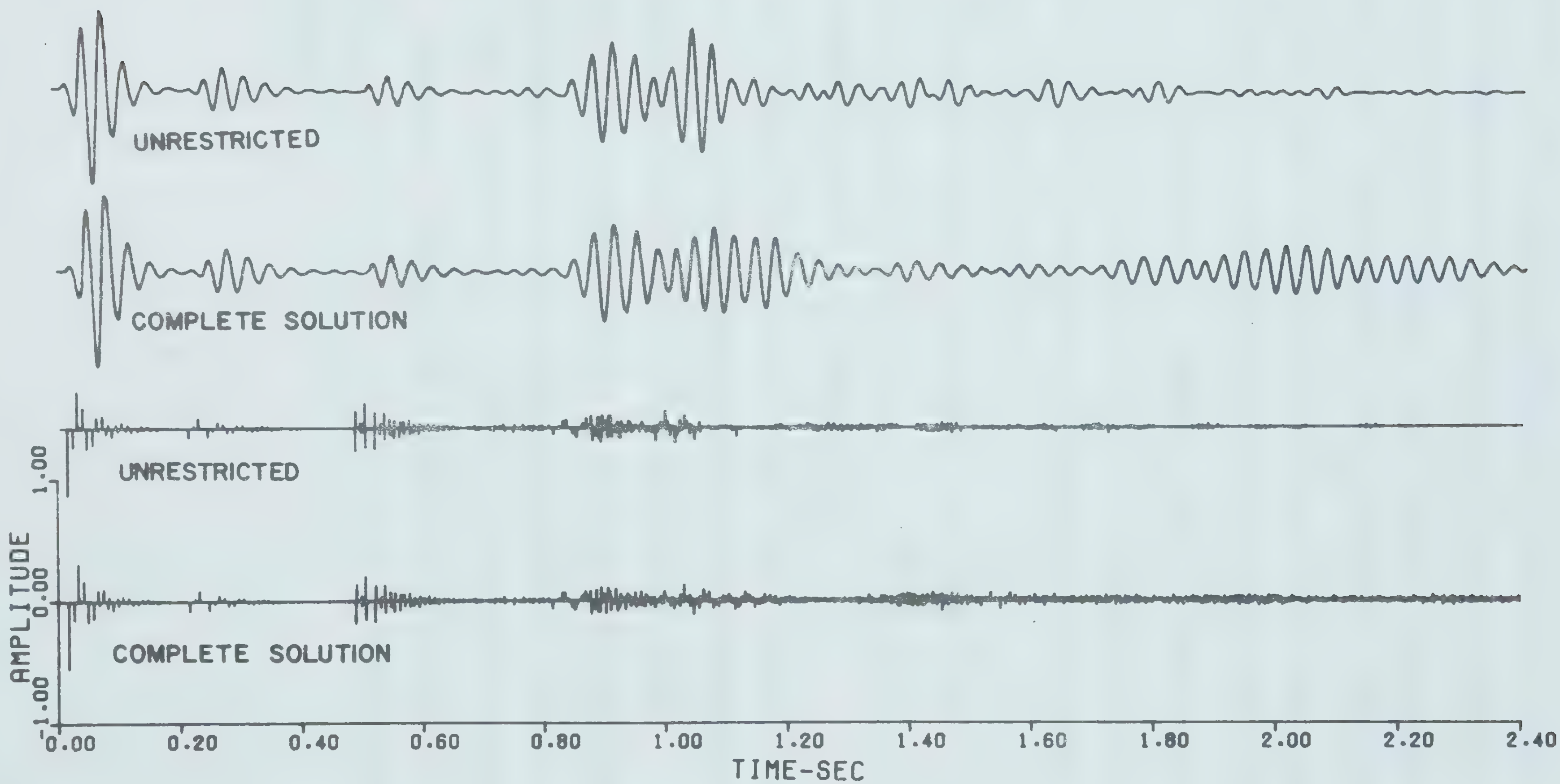
the Unrestricted sequence in Figure 12. These are shown in Figure 21 together with the synthetic seismograms. As can be seen, within the time window of 1.105 seconds the agreement is very good. This relieves any doubt which may have existed as to how meaningful previous comparisons have been.

3.6 Conclusions

The new approach developed for the generation of the ray code using the classification scheme of Hron (1972) proved to be successful in reducing computer storage costs. For the sixteen layer model with HMS = LAY, the reduction in total cost for generating the synthetic seismogram was slightly greater than one-half. Restricted partitioning further reduced costs by generally suppressing those contributions outside the time range of the velocity structure. The Severity Level 1 restriction resulted in a synthetic seismogram at approximately 1/20th the cost required to produce the Unrestricted sequence using the code generation scheme described in Chapter I. In addition the Severity Level 1 sequence retained most of the character observed in the Unrestricted sequence within the time window of 1.105 seconds which marks the end of the velocity structure in two-way travel time.

Inclusion of the geometric spreading correction in the amplitude computation was found to significantly affect the character of the synthetic seismograms, particularly

Figure 21. Comparison of the Unrestricted and Complete Wave Solutions for the 16-Layer Model



the relative amplitude of different groups of arrivals.

In comparing the method of n^{th} order multiples with the classification scheme of Hron (1972), it was found that for the sixteen layer model, the synthetics including primaries plus first and second order multiples compared closely with the Severity Level 3 sequence within the time window of 1.105 seconds. However the n^{th} order multiple approach seems to be less selective in the rays that it generates in the sense that a large majority of these rays fall outside the time window.

N^{th} order surface multiples were found to be a good approximation to the complete set of n^{th} order multiples for the model used in the comparison.

The Unrestricted sequence for the sixteen layer model was seen to correspond very well with the complete wave solution thereby justifying previous comparisons.

Some time after completing the work on the permuted partition approach it was discovered that a similar technique had been successfully applied by Vered and Ben-Menahem (1974). However, they do not appear to make use of restricted partitioning to reduce the number of undesirable phases.

BIBLIOGRAPHY

Hron, F., Numerical Methods of Ray Generation in Multi-Layered Media. In "Methods in Computational Physics, Vol. 12: Theoretical Seismology", B.A. Bolt, ed., Academic Press, New York and London, 1972.

Hron, M.P., Waves in Inhomogeneous Media, Ph.D. Thesis, University of Alberta, Edmonton, 1973.

Johnson, S.M., Generation of Permutations by Adjacent Transposition, RM-3177-PR, The RAND Corporation, Santa Monica, CA, 1962; Math. Comp. 17 (1963), 282-285.

Lehmer, D.H., The Machine Tools of Combinatorics. In "Applied Combinatorial Mathematics", E.F. Beckenbach, ed., John Wiley and Sons, Inc., New York, London, and Sydney, 1964.

Paige, L.J., and C.B. Tompson, The Size of the 10 x 10 Orthogonal Latin Square Problem, Chapter 5 in "Combinational Analysis, Proc. Sympos. Appl. Math., 10, 71-83, American Mathematical Society, Providence, Rhode Island, 1960.

Riordan, J., An Introduction to Combinatorial Analysis, John Wiley and Sons, Inc., New York, London, and Sydney, 1958.

Vered, M., and A. Ben-Menahem, Applications of Synthetic Seismograms to the Study of Low-Magnitude Earthquakes and Crustal Structure in the Northern Red Sea Region, B.S.S.A., 64, 1221-1237, 1974.

Watson, R.J., Decomposition and Suppression of Multiple Reflections, Geophysics, 30, 54-71, 1965.

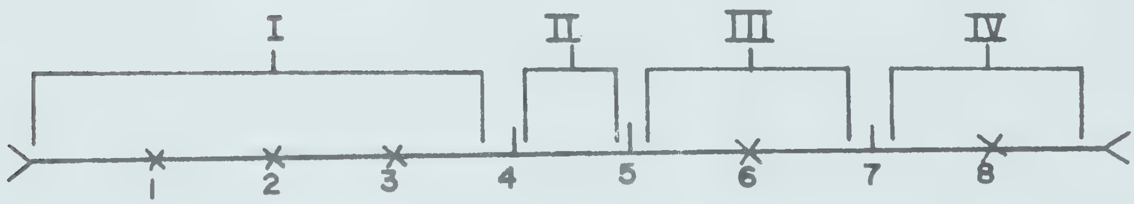
Wells, M.B., Generation of Permutations by Transposition, Math. Comp., 15 (1961), 192-195.

APPENDIX A

NUMBER OF ARRANGEMENTS OF n_2 IDENTICAL BALLS INTO n_1 POCKETS, SOME OF WHICH CAN BE EMPTY (FROM HRON (1972))

Consider an arbitrary distribution of n_2 identical balls into $n_1 \geq 1$ pockets separated by $(n_1 - 1)$ inner walls. Each such distribution can be visualized as a sequence of $(n_1 - 1 + n_2)$ items represented by n_2 balls and $(n_1 - 1)$ inner boundaries, as illustrated in Figure A.1 for $n_1 = 4$, $n_2 = 5$. From this diagram, we can see that three inner

Figure A.1. One Distribution of 5 Balls into 4 Pockets.



walls occupy, in the sequence of eight elements, the fourth, fifth, and seventh positions so that there are 3 balls in the first pocket, the second pocket is left empty, and there is 1 ball in pockets III and IV.

Because individual distributions differ from each

other only in selection of $(n_1 - 1)$ positions occupied by inner boundaries, the number, M_1, of distributions of n_2 balls into n_1 pockets, some of which can be left empty, is equal to the number of all possible selections of $(n_1 - 1)$ positions from the total number $(n_1 - 1 + n_2)$. Obviously, this is equal to

$$M_1(n_1,n_2) = C_{n_1-1}^{n_1+n_2-1} = C_{n_2}^{n_1+n_2-1} = \frac{(n_1 + n_2 - 1)!}{(n_1 - 1)!n_2!} . \quad \text{A.1.1}$$

There $C_{n_1-1}^{n_1+n_2-1}$ is the number of all combinations of the $(n_1 - 1)$th class from $(n_1 + n_2 - 1)$ elements.

APPENDIX B

NUMBER OF ARRANGEMENTS OF n_2 IDENTICAL BALLS INTO n_1 POCKETS, NONE OF WHICH CAN BE EMPTY (FROM HRON (1972))

Distributions of this kind can be performed only if $n_2 \geq n_1$. Then the problem can be transformed easily to the problem resolved in the previous section if we separate from n_2 balls n_1 so-called "fixed balls" by distributing them into n_1 pockets, none of them leaving empty. Then the number M_2 has to be equal to the number of arrangements of the remaining $(n_2 - n_1)$ "movable balls" into n_1 pockets provided that some of them may not contain any movable ball. Using Equation A.1.1 we get immediately

$$M_2(n_1,n_2) = M_1(n_1,n_2 - n_1) = C_{n_1-1}^{n_2-1} = \frac{(n_2 - 1)!}{(n_1 - 1)!(n_2 - n_1)!} \qquad \text{B.1.1}$$

where $M_2(n_1,n_2)$ is the number of all arrangements of n_2 identical balls into n_1 pockets leaving none of them empty.

APPENDIX C

COMPUTER PROGRAMS

C.1 N-th Order Multiples Approach

```
  3     FORMAT(//,T10,'DOWNWARD REFLECTION COEFFICIENT',//)
 12     FORMAT(7F10.5)
  4     FORMAT(//,T10,'DOWNWARD TRANSMISSION COEFFICIENT',//)
 13     FORMAT(7F10.5)
  5     FORMAT(//,T10,'UPWARD REFLECTION COEFFICIENT',//)
 14     FORMAT(7F10.5)
  6     FORMAT(//,T10,'UPWARD TRANSMISSION COEFFICIENT',//)
 15     FORMAT(7F10.5)
  7     FORMAT('TIME=',F10.5)
  8     FORMAT('AMP=',E15.4)
 11     FORMAT(//,T28,'PRIMARY REFLECTIONS',//)
 16     FORMAT(//,'GROUP SYMBOL',T30,I5)
 17     FORMAT(T40,I5)
 21     FORMAT(4F10.5)
 22     FORMAT(5F10.2)
 23     FORMAT(F10.2)
504     FORMAT(/,'TOTAL NUMBER OF PHASES',T30,I10)
        DIMENSION THICK(16)
        DIMENSION VEL(17)
        DIMENSION DEPTH(16)
        DIMENSION TIME(16)
C VALUE CONTAINS THE  WAVELET PLUS TRAILING
C ZEROS. THE DIMENSION OF VALUE MUST BE GREATER THAN OR
C EQUAL TO THE LENGTH OF THE CONVOLUTION OF THE WAVELET
C WITH THE SPIKE SYNTHETIC, WITH THE ADDED CONDITION THAT
C THE DIMENSION MUST EQUAL 2**N WHERE N IS SOME POSITIVE
C INTERGER.    2**5=32  2**6=64  2**7=128  2**8=256  2**9=512
C 2**10=1024  2**11=2048  2**12=4096  2**13=8192  2**14=16384
        COMPLEX VALUE(8192)
C P50 CONTAINS THE SPIKE SYNTHETIC(OUTPUT FROM RDOFF)
C DIMENSION P50(SAME DIMENSION AS VALUE)
        COMPLEX P50(8192)
        DIMENSION P60(4602)
C DIMENSION BNUIS(2 * DIMENSION OF VALUE)
        DIMENSION BNUIS(16384)
        EQUIVALENCE(VALUE,BNUIS)
C C CONTAINS THE Y OR AMPLITUDE VALUES USED IN PLOTING THE
C SYNTHETIC SEISMOGRAM
C DIMENSION C(LENGTH OF THE CONVOLUTION OF THE SPIKE SYNTHETIC
C WITH THE WAVELET  +  2) THE SCALING PARAMETERS ARE STORED IN
C THE LAST TWO POSITIONS
        DIMENSION C(4602)
C C201 CONTAINS THE X OR TIME VALUES USED IN PLOTTING THE SS
C DIMENSION C201(SAME AS DIMENSION OF C)
        DIMENSION C201(4602)
C IN NLOGN THE DATA LENGTH MUST BE EQUAL TO 2**N WHERE N IS
C SOME POSITIVF INTERGER. DIMENSION MMM(LARGEST VALUE OF N TO
C BE PROCESSED)
        DIMENSION MMM(13)
C GTIME(I) REPRESENTS THE VERTICAL TRAVEL TIME FROM THE SURFACE
```

```
C TO THE BOTTOM OF THE ITH LAYER
      DIMENSION GTIME(131)
      COMPLEX B,CMPLX

      DO 45 I=1,8192
      P50(I)=CMPLX(0.0,0.0)
   45 CONTINUE

      LAY=16
      JJ50=LAY+1
C INPUT DATA
      READ(7,23)  (DEPTH(I),I=1,LAY)
      READ(8,23)  (VEL(I),I=1,JJ50)
C IF THE GEOMETRICAL SPREADING FACTOR IN THE AMPLITUDE IS NOT
C DESIRED SPECIFY IIG=0. ANY OTHER VALUE FOR IIG
C WILL INVOKE THE SPREADING CALCULATION.  BE CERTAIN
C TO CHANGE THE VERTICAL SCALING PARAMETERS IN THE
C PLOTTING ROUTINE WHEN RUNNING THIS PROGRAM
C WITHOUT GEOMETRICAL SPREADING
      IIG=0
C IF FIRST ORDER MULTIPLES ARE NOT DESIRED
C SPECIFY IIF=0
      IIF=1
C IF SECOND ORDER MULTIPLES ARE NOT DESIRED
C SPECIFY IIS=0
      IIS=1
C TO SUPPRESS PRINTOUT OF REFLECTION AND TRANSMISSION
C COEFFICIENTS SPECIFY ISUPCO=0.  ALL OTHER VALUES
C WILL RESULT IN A PRINTOUT
      ISUPCO=0
C TO SUPPRESS PRINTOUT OF AMPLITUDES AND TIMES
C FOR DIRECT REFLECTIONS SPECIFY ISUPP=0.  ALL
C OTHER VALUES WILL RESULT IN A PRINTOUT
      ISUPP=0
C TO SUPPRESS PRINTOUT OF AMPLITUDE AND TIMES FOR
C FIRST ORDER MULTIPLES SPECIFY ISUPFO=0.  ALL OTHER
C VALUES WILL RESULT IN A PRINTOUT
      ISUPFO=0
C TO SUPPRESS PRINTOUT OF AMPLITUD AND TIMES FOR
C SECOND ORDER MULTIPLES SPECIFY ISUPSO=0.  ALL OTHER
C VALUES WILL RESULT IN A PRINTOUT
      ISUPSO=0
C TO SUPPRESS PLOT SPECIFY ISUPLOT=0.  ALL OTHER
C VALUES WILL RESULT IN A PLOT OF THE SYNTHETIC
      ISUPLT=1
      THICK(1)=DEPTH(1)
      M=LAY-1
      M1=LAY-2

      DO 10 I=2,LAY
      THICK(I) = DEPTH(I-DEPTH(I-1)
   10 CONTINUE

      DO 20 I=1,LAY
```

```
      TIME(I) = THICK(I)/VEL(I)
 20   CONTINUE

C     CALCULATE THE REFLECTION AND TRANSMISSION COEFFICIENTS
      DIMENSION DREFCO(16)
      DIMENSION DTRACO(16)
      DIMENSION UREFCO(16)
      DIMENSION UTRACO(16)

      DO 25 I=1,LAY
      DREFCO(I)=(VEL(I-VEL(I+1))/(VEL(I) + VEL(I+1))
 25   CONTINUE

      IF(ISUPCO .EQ. 0) GO TO 49
C     WRITE(6,3)
C     WRITE(6,21) (DREFCO(I),I=1,LAY)

 49   DO 30 I=1,LAY
      DTRACO(I)=1+DREFCO(I)
  30  CONTINUE

      IF(ISUPCO.EQ.0) GO TO 46
C     WRITE(6,4)
C     WRITE(6,21) (DTRACO(I),I=1,LAY)
 46   UREFCO(1)=1.0

      DO 35 I=2,LAY
      UREFCO(I)=(VEL(I-VEL(I-1))/(VEL(I-1)+VEL(I))
C      UREFCO INDICATES REFL. FR. TOP OF ITH LAYER
  35  CONTINUE

      IF(ISUPCO.EQ.0) GO TO 47
C     WRITE(6,5)
C     WRITE(6,21) (UREFCO(I),I=1,LAY)

 47   DO 40 I=2,LAY
      UTRACO(I)=1+UREFCO(I)
  40  CONTINUE

      UTRACO(1)=1.0
      IF(ISUPCO.EQ.0) GO TO 48
      WRITE(6,6)
      WRITE(6,21) (UTRACO(I),I=1,LAY)
 48   IAFG=0
C      NOW CALCULATE TIME AND AMPLITUDE FOR DIRECT REFLECTIONS
      AMP=1.0
      IF(ISUPP.EQ.0) GO TO 43
      WRITE(6,11)
      IND=1
 43   TIM=2*TIME(1)
      IF(IIG .EQ. 0) GO TO 41
      AMP=AMP*DREFCO(1)/(2*DEPTH(1))

      GO TO 42
```

```
 41     AMP=AMP*DREFCO(1)
 42     IF(ISUPP.EQ.0) GO TO 44
        WRITE(6,16) IND
        WRITE(6,17) IND
        WRITE(6,8) AMP
        WRITE(6,7) TIM
 44     B=CMPLX(AMP,TIM)
        CALL RDOFF(B,P50,IAFG)
        GTIME(1)=TIME(1)

        DO 80 I=2,LAY
        AMP=1.0
        I1=I-1

        DO 65 J=1,I1
        AMP=AMP*DTRACO(J)
 65     CONTINUE

        DO 75 J=1,I
        AMP=AMP*UTRACO(J)
 75     CONTINUE

        TIM=TIM+2*TIME(I)
        IND=IND+1
        IF(IIG .EQ. 0) GO TO 76
        PL=0.
        DO 78 J=1,I
        PL=PL+2*THICK(I)/VEL(1)
 78     CONTINUE
        AMP=AMP*DREFCO(I)/PL

        GO TO 77

 76     AMP=AMP*DREFCO(I)
 77     GTIME(I)=TIM/2
        B=CMPLX(AMP,TIM)
        IF(ISUPP.EQ.0) GO TO 86
        WRITE(6,16) IND
        WRITE(6,17) IND
        WRITE(6,8) AMP
        WRITE(6,7) TIM
 86     CALL RDOFF(B,P50,IAFG)
 80     CONTINUE

        IF(IIF.EQ.0) GO TO 81
        CALL FOM(IND,LAY,TIME,GTIME,DREFCO,THICK,DEPTH,IIG,
      CDTRACO,UREFCO,UTRACO,ISUPFO,B,P50,IAFG)
 81     IF(IIS.EQ.0) GO TO 82
        CALL SOM(IND,LAY,TIME,GTIME,DREFCO,THICK,DEPTH,IIG,
      CDTRACO,UREFCO,UTRACO,ISUPSO,B,P50,IAFG)
C GENERATE THE TIME AXIS FOR PLOTTING
        IF(ISUPLT.EQ.0) GO TO 2004
 82     C201(1)=0.0
```

```
C INSERT DO LOOP PARAMATER. I=2,LENGTH OF TRACE
C IN SECONDS X 1000

      DO 2001 I=1,4600
      C201(I)=I/1000.
 2001 CONTINUE

C SPECIFY THE KLAUDER WAVELET PARAMETERS
C     CALL KLAUD(7.,12.,4.,VALUE)
      CALL WAVLT1(VALUE)
      DO 83 I=1,4602
      P60(I)=REAL(P50(I))
   83 CONTINUE
      WRITE(11) P60
C IN THE NEXT TWO STATEMENTS THE FIRST PARAMETER OF
C NLOGN MUST EQUAL THE DIMENSION OF MMM( )
      CALL NLOGN(13,VALUE,-1.0)
      CALL NLOGN(13,P50,-1.0)

      DO 2002 I=1,8192
      VALUE(I)=VALUE(I)*P50(I)
 2002 CONTINUE

C INSERT 1ST NLOGN PARAMETER
      CALL NLOGN(13,VALUE,1.0)
C INSERT DO LOOP PARAMETER, I=1,LENGTH OF TRACE
C IN SECONDS X 1000

      DO 2003 I=1,4600
      C(I)=BNUIS(2*I-1)
 2003 CONTINUE

      WRITE(6,504) IAFG
      CALL PSY(C,C201,P60)
 2004 STOP
      END


C 'FOM' CALCULATES THE TIME ARRIVAL TIM AND AMPLITUDE
C OF THE FIRST ORDER MULTIPLES
      SUBROUTINE FOM(IND,LAY,TIME,GTIME,DREFCO,THICK,DEPTH,IIG,
     CDTRACO,UREFCO,UTRACO,ISUPFO,B,P50,IAFG)
      DIMENSION GTIME(131)
      COMPLEX B,CMPLX
      COMPLEX P50(8192)
      DIMENSION DREFCO(131)
      DIMENSION DTRACO(131)
      DIMENSION UREFCO(131)
      DIMENSION UTRACO(131)
      DIMENSION TIME(131)
      DIMENSION THICK(131)
      DIMENSION DEPTH(131)
  100 FORMAT(//,T28,'FIRST ORDER MULTIPLES',//)
  101 FORMAT(//,'GROUP NUMBER',T30,I5)
```

```
102     FORMAT('+',T40,3I4)
103     FORMAT('AMP =', E15.4)
104     FORMAT('TIME =',F10.5)
105     FORMAT(F10.5)
        AMP3=1.0
        IF(ISUPFO.EQ.0) GO TO 110
        WRITE(6,100)

110     DO 200 IA=1,LAY
        PTIME=GTIME(IA)
        GSP=0.
        DO 210 I=1,IA
        GSP=GSP+THICK(I)*VEL(I)/VEL(1)
210     CONTINUE
        AMPA = DREFCO(IA)*AMP3
        AMP2=1.0
        PTIME1=0.0
        PTIME6=GTIME(IA)
        GSP1=0.0
        GSP6=GSP

        DO 180 IB=1,IA
        PTIME1=PTIME1+PTIME6
        PTIME4=PTIME1+PTIME
        GSP1=GSP1+GSP6
        GSP4=GSP1+GSP
        AMPA=AMPA*AMP2*UREFCO(IB)
        AMP1=1.0
        PTIME2=0.0
        PTIME5=TIME(IB)
        GSP2=0.0
        GSP5=THICK(IB)*VEL(IB)/VEL(1)

        DO 150 IC=IB,LAY
        PTIME2=PTIME2+PTIME5
        GSP2=GSP2+GSP5
        GG=0.
        DO 220 I=1,IC
        GG=THICK(I)*VEL(I)/VEL(1)
220     CONTINUE
        GSPF=GG+GSP2+GSP4
        AMP1=DREFCO(IC)*AMP1
        IF(IIG .EQ. 0) GO TO 107
        AFOM=AMPA*AMP1/GSPF

        GO TO 108

107     AFOM=AMPA*AMP1
108     TFOM=GTIME(IC)+PTIME2+PTIME4
        IND=IND+1
        B=CMPLX(AFOM,TFOM)
        IF(ISUPFO.EQ.0) GO TO 111
        WRITE(6,101) IND
111     IBM1=IB-1
```

```
      IF(ISUPFO.EQ.0) GO TO 112
      WRITE(6,102) IA, IBM1,IC
      WRITE(6,103) AFOM
      WRITE(6,104) TFOM
 112  CALL RDOFF(B,P50,IAFG)
      ICP1=IC+1
      IF(ICP1.GT.LAY) GO TO 150
      AMP1=DTRACO(IC)*UTRACO(ICP1)*AMP1/DREFCO(IC)
      PTIME5=TIME(ICP1)
      GSP5=THICK(ICP1)*VEL(ICP1)/VEL(1)
 150  CONTINUE

      AMP2=1/UTRACO(IB)
      PTIME6=-TIME(IB)
      GSP6=-THICK(IB)*VEL(IB)/VEL(1)
 180  CONTINUE

      IAP1=IA+1
      IF(IAP1 .GT.LAY) GO TO 200
      AMP3=DTRACO(IA)*UTRACO(IAP1)*AMP3
 200  CONTINUE

      RETURN
      END


C 'SOM' CALCULATES THE ARRIVAL TIME AND AMPLITUDE
C OF THE SECCND ORDER MULTIPLES
      SUBROUTINE SOM(IND,LAY,TIME,GTIMF,DREFCO,THICK,DEPTH,IIG,
     CDTRACO,UREFCO,UTRACO,ISUPSO,B,P50,IAFG)
      COMPLEX P50(8192)
      DIMENSION GTIME(131)
      COMPLEX B,CMPLX
      DIMENSION DREFCO(131)
      DIMENSION DTRACO(131)
      DIMENSION UREFCO(131)
      DIMENSION UTRACO(131)
      DIMENSION TIME(131)
      DIMENSION DEPTH(131)
      DIMENSION THICK(131)
 100  FORMAT(//,T28,'SECOND ORDER MULTIPLES',//)
 101  FORMAT(//,'GROUP SYMBOL',T30,I5)
 102  FORMAT('+',T40,5I4)
 103  FORMAT('AMP =',E15.5)
 104  FORMAT('TIME =',F10.5)
 105  FORMAT(F10.5)
 106  FORMAT('GSPF =',E15.4)
      IF(ISUPSO.EQ.0) GO TO 110
      WRITE(6,100)
 110  AMP10=1.0

      DO 200 IA=1,LAY
      PTIME=GTIME(IA)
      GSP=0.
```

```
      DO 210 I=1,IA
      GSP=GSP+THICK(I)*VEL(I)/VEL(1)
210   CONTINUE
      AMPA=DREFCO(IA)*AMP10
      AMP9=1.0
      AMP17=AMPA
      PTIME1=0.0
      PTIME6=GTIME(IA)
      GSP1=0.0
      GSP6=GSP

      DO 180 IB=1,IA
      PTIME1=PTIME1+PTIME6
      PTIME4=PTIME1+PTIME
      GSP1=GSP1+GSP6
      GSP4=GSP1+GSP
      AMPA=AMP17*AMP9*UREFCO(IB)
      AMP8=1.0
      AMP15=AMPA
      PTIME2=0.0
      PTIME5=TIME(IB)
      GSP2=0.0
      GSP5=THICK(IB)*VEL(IB)/VEL(1)

      DO 150 IC=IB,LAY
      PTIME2=PTIME2+PTIME5
      GSP2=GSP2+GSP5
      AMP8=DREFCO(IC)*AMP8
      AMP7=1.0
      AMPA=AMP15*AMP8
      AMP16=AMPA
      PTIME3=0.0
      PTIME7=GTIME(IC)
      GSP7=0.0
      GSP8=0.
      DO 220 I=1,IC
      GSP8=GSP8+THICK(I)*VEL( I)/VEL(1)
 220  CONTINUE
      DO 140 ID=1,IC
      PTIME3=PTIME3+PTIME7
      GSP7=GSP7+GSP8
      AMPA=AMP16*AMP7*UREFCO(ID)
      AMP6=1.0
      PTIME8=0.0
      PTIME9=TIME(ID)
      GSP9=0.0
      GSP10=THICK(ID)*VEL(ID)/VEL(1)
      DO 130 IE=ID,LAY
      PTIME8=PTIME8+PTIME9
      GSP9=GSP9+GSP10
      AMP6=AMP6*DREFCO(IE)
      GPPR=0.
      DO 230 I=1,IE
      GPPR=GPPR+THICK(I)*VEL(I)/VEL(1)
```

```
 230    CONTINUE
        GSPF=GPPR+GSP9+GSP7+GSP2+GSP4
        IF(IIG .EQ.0) GO TO 107
        AFOM=AMPA*AMP6/GSPF
        GO TO 108
 107    AFOM=AMPA*AMP6
 108    TFOM=GTIME(IE)+PTIME2+PTIME4+PTIME3+PTIME8
        IND=IND+1
        B=CMPLX(AFOM,TFOM)
        IF(ISUPSO.EQ.0) GO TO 111
        WRITE(6,101) IND
 111    IBM1=IB-1
        IDM1=ID-1
        IF(ISUPSO.EQ.0) GO TO 112
        WRITE(6,102) IA,IBM1,IC,IDM1,IE
        WRITE(6,103) AFOM
        WRITE(6,104) TFOM
 112    CALL RDOFF(B,P50,IAFG)
        IEP1=IE+1
        IF(IEP1.GT.LAY) GO TO 130
        AMP6=UTRACO(IEP1)*DTRACO(IE)*AMP6/DREFCO(IE)
        PTIME9=TIME(IEP1)
        GSP10=THICK(IEP1)*VEL(IEP1)/VEL(1)
 130    CONTINUE
        AMP7=1.0/UTRACO(ID)
        PTIME7=-TIME(ID)
        GSP8=-THICK(ID)*VEL(ID)/VEL(1)
 140    CONTINUE
        ICP1=IC+1
        IF(ICP1.GT.LAY) GO TO 150
        AMP8=UTRACO(ICP1)*DTRACO(IC)*AMP8/DREFCO(IC)
        PTIME5=TIME(ICP1)
        GSP5=THICK(ICP1)*VEL(ICP1)/VEL(1)
 150    CONTINUE
        AMP9=1.0/UTRACO(IB)
        PTIME6=-TIME(IB)
        GSP6=-THICK(IB)*VEL(IB)/VEL(1)
 180    CONTINUE
        IAP1=IA+1
        IF(IAP1 .GT.LAY) GO TO 200
        AMP10=DTRACO(IA)*UTRACO(IA+1)*AMP10
 200    CONTINUE
        RETURN
        END


C 'PSY' PLOTS THE SYNTHETIC SEISMOGRAM
        SUBROUTINE PSY(C,C201,P60)
C DIMENSION C( -SEE MAIN
        DIMENSION C(4602)
C DIMENSION C201( -SEE MAIN
        DIMENSION C201(4602)
        DIMENSION P60(1)
        CALL PLOTS
```

```
C MOVE ORIGIN IN TO ALLOW FOR LABELS ETC.
      CALL PLOT(8.0,8.0,-3)
C FOR THE TIME VALUES FORCE 10 INCHES PER SEC.
      XMIN=0.0
      XDELT=.2
C STORE SCALING PARAMETERS FOR THE TIME AXIS
      C201(4601)=XMIN
      C201(4602)=XDELT
C FOR THE AMPLITUDE AXIS WILL FORCE 1./INCH
      YMIN=0.0
      YDELT=1.
C STORE SCALING PARAMETERS FOR AMPLITUDE AXIS
      C(4601)=YMIN
      C(4602)=YDELT
C PLOT TIME AXIS
      CALL AXIS(0.0,-1.0,'TIME-SEC',-8,15.0,0.0,0.0,.2)
C PLOT AMPLITUDE AXIS
      CALL AXIS(0.0,-1.0,'AMPLITUDE',9,2.0,90.0,-1.,1.)
C PLOT DATA
      CALL PLOT(0.0,0.0,3)
      DO 1 I=1,3000
      XP=(C201(I-C201(4601))/C201(4602)
      TP=C(I)/C(4602)
      CALL PLOT(XP,TP,2)
  1   CONTINUE
C PLOT SPIKE SYNTHETIC
C MOVE ORIGIN UP
      CALL PLOT(8.0,16.0,-3)
C PLOT TIME AXIS
      CALL AXIS(0.0,-1.0,'TIME-SEC',-8,15.0,0.0,0.0,.2)
C PLOT AMPLITUDE AXIS
      CALL AXIS(0.0,-1.0,'AMPLITUDE',9,2.0,90.0,-1.,1.)
C PLOT DATA FOR SPIKE SYNTHETIC
      CALL PLOT(0.0,0.0,3)
      DO 2 I=1,3000
      XP=(C201(I-C201(4601))/C201(4602)
      TP=P60(I)/YDELT
      CALL PLOT(XP,0.0,2)
      CALL PLOT(XP,TP,2)
      CALL PLOT(XP,0.0,2)
 2    CONTINUE
      CALL PLOT(2.0,2.0,999)
      RETURN
      END


C 'RDOFF' ROUNDS OFF THE TIME OF ARRIVAL OF A SEISMIC PULSE
C TO 3 DECIMAL DIGITS AND PLACES THE AMPLITUDE VALUE IN
C P50(IDIV) WHERE IDIV/1000 IS THE TIME OF ARRIVAL TO
C THE CLOSEST MILLISECOND
      SUBROUTINE RDOFF(B,P50,IAFG)
      COMPLEX B
C DIMENSION P50( -SEE MAIN
      COMPLEX P50(8192)
```

```
      IAFG=IAFG+1
      TIME1=(AIMAG(B))*1000.
      TIME2=TIME1+.5
      IDIV=TIME2
      P50(IDIV)=CMPLX(REAL(P50(IDIV))+REAL(B),0.0)
      RETURN
      END


      SUBROUTINE NLOGN(N,X,SIGN)
C DIMENSION MMM( -SEE MAIN
      DIMENSION MMM(13)
C DIMENSION X( -SEE MAIN
      DIMENSION X(8192)
      COMPLEX X,WK,HOLD,Q
      LX=2**N
      DO 1 I=1,N
    1 MMM(I)=2**(N-I)
      DO 4 L=1,N
      NBLOCK=2**(L-1)
      LBLOCK=LX/NBLOCK
      LBHALF=LBLOCK/2
      K=0
      DO 4 IBLOCK=1,NBLOCK
      FK=K
      FLX=LX
      V=SIGN*6.2831853*FK/FLX
      WK=CMPLX(COS(V),SIN(V))
      ISTART=LBLOCK*(IBLOCK-1)
      DO 2 I=1,LBHALF
      J=ISTART+I
      JH=J+LBHALF
      Q=X(JH)*WK
      X(JH)=X(J-Q
      X(J)=X(J)+Q
    2 CONTINUE
      DO 3 I=2,N
      II=I
      IF(K.LT.MMM(I)) GO TO 4
    3 K=K-MMM(I)
    4 K=K+MMM(II)
      K=0
      DO 7 J=1,LX
      IF(K.LT.J) GO TO 5
      HOLD=X(J)
      X(J)=X(K+1)
      X(K+1)=HOLD
    5 DO 6 I=1,N
      II=I
      IF(K.LT.MMM(I)) GO TO 7
    6 K=K-MMM(I)
    7 K=K+MMM(II)
      IF(SIGN.LT.0.0) RETURN
      DO 8 I=1,LX
```

```
    8   X(I)=X(I)/FLX
        RETURN
        END

C SUBROUTINE KLAUD GENERATES A NORMALIZED KLAUDER WAVELET
C F1 - THE TERMINAL LOW FREQUENCY
C IZ - THE FREQUENCY BANDWIDTH IN OCTAVES
C T - THE DURATION OF THE SIGNAL IN SECONDS
C THE GENERATION IS DONE USING EQN.7 S.S.C. TRAINING DEPT.
C CATALOG OF KLAUDER WAVELETS
        SUBROUTINE KLAUD (T,F1,Z,VALUE)
C DIMENSION VALUE( ) IN THE FOLLOWING STATEMENT-SEE MAIN
        COMPLEX VALUE(8192), CMPLX
        PI=3.141592
C 'VALUE(301) IN THE NEXT STATEMENT REPRESENTS THE AUTO-
C CORRELATION OF THE SWEEP AT ZERO SHIFT AND SHOULD
C THEORETICALY HAVE A VALUE OF'T', WHERE 'T' IS THE
C TIME LENGTH OF THE SWEEP. IF NORMALIZATION IS
C DESIRED SPECIFY 'COMPLEX(1.0,0.0)' RATHER  THAN
C COMPLEX(T,0.0)IN THE NEXT STATEMENT
        VALUE(301)=CMPLX(1.0,0.0)
        I=301
        SHIFT=.001
        DO 16 J=1,300
        A=PI*F1*SHIFT*((2**Z-1)*(1-SHIFT/T)
        B=PI*F1*SHIFT*((2**Z-1)
        C=PI*F1*SHIFT*((2**Z)+1)
        VALUE(I+J)=CMPLX(1.0*SIN(A)*COS(C)/B,0.0)
        VALUE(I-J)=VALUE(I+J)
        SHIFT=SHIFT+.001
   16   CONTINUE
        DO 18 I=602,8192
        VALUE(I)=CMPLX(0.0,0.0)
   18   CONTINUE
        RETURN
        END

C SUBROUTINE WAVLT1 STORES IMP. LOG WAVELET
C IN VALUE
        SUBROUTINE WAVLT1(VALUE)
        COMPLEX VALUE(8192),CMPLX
        DIMENSION PALL(97)
   10   FORMAT(5F10.5)
        READ(13,10)  (PALL(I),I=1,97)
        DO 15 I=1,97
        VALUE(I)=CMPLX(PALL(I),0.0)
   15   CONTINUE
        DO 20 I=98,8192
        VALUE(I)=CMPLX(0.0,0.0)
   20   CONTINUE
        RETURN
        END
```

C.2 The Permuted Partition Approach

```
C 'FACT' IS CALLED BY 'NODYAN' AND COMPUTES NJ FACTORIAL
      SUBROUTINE FACT(AJ,AJF)
      IF (AJ .EQ. 0.0) GO TO 14
      IF (AJ .EQ. 1.0) GO TO 14
      AJM2=AJ-2.0
      NJM2=AJM2
      IF(AJM2 .EQ. 0.0) GO TO 15
      AJF=AJ

      DO 13 I=1,NJM2
      BJ=AJ-I
      AJF=AJF*BJ
 13   CONTINUE

      GO TO 16
  14  AJF=1.0
      GO TO 16
  15  AJF=AJ
  16  RETURN
      END


C 'NODYAN CALCULATES THE NUMBER OF DYNAMIC ANALOGS
C ASSOCIATED WITH A PARTICULAR RAY CODE
C REFERENCE EQN. 19 HRON 1972
      SUBROUTINE NODYAN(MAC,ICN,ICM,AF)
      DIMENSION ICN(17)
      DIMENSION ICM(16)
      IC=MAC-1
      AF=1.0

      DO 18 I=1,IC
      A15=ICN(I)
      CALL FACT(A15,AJF)
      A20=ICM(I)
      CALL FACT(A20,AJG)
      CALL FACT((A15-A20),AJH)
      AZ1=AJF/(AJG*AJH)
      IP1=I+1
      A30=ICN(IP1)
      CALL FACT((A30-1.0),AJI)
      A60=A30-A15+A20
      CALL FACT(A60,AJJ)
      A50=A15-A20-1.0
      CALL FACT((A50),AJK)
      AZ2=AJI/(AJJ*AJK)
      ANC=AZ1*AZ2
      AF=AF*ANC
 18   CONTINUE
```

```
      RETURN
      END


C 'RDOFF' ROUNDS OFF THE TIME OF ARRIVAL OF A SEISMIC PULSE
C TO 3 DECIMAL DIGITS AND PLACES THE AMPLITUDE VALUE IN
C P50(IDIV) WHERE IDIV/1000 IS THE TIME OF ARRIVAL TO
C THE CLOSEST MILLISECOND
      SUBROUTINE RDOFF(B,P50)
      COMPLEX B
      COMPLEX P50(8192)
      TIME1=(AIMAG(B))*1000.
      TIME2=TIME1+.5
      IDIV=TIME2
      P50(IDIV)=CMPLX(REAL(P50(IDIV))+REAL(B),0.0)
      RETURN
      END


      SUBROUTINE NLOGN(N,X,SIGN)
      DIMENSION MMM(13)
      DIMENSION X(8192)
      COMPLEX X,WK,HOLD,Q
      LX=2**N

      DO 1 I=1,N
    1 MMM(I)=2**(N-I)

      DO 4 L=1,N
      NBLOCK=2**(L-1)
      LBLOCK=LX/NBLOCK
      LBHALF=LBLOCK/2
      K=0

      DO 4 IBLOCK=1,NBLOCK
      FK=K
      FLX=LX
      V=SIGN*6.2831853*FK/FLX
      WK=CMPLX(COS(V),SIN(V))
      ISTART=LBLOCK*(IBLOCK-1)

      DO 2 I=1,LBHALF
      J=ISTART+I
      JH=J+LBHALF
      Q=X(JH)*WK
      X(JH)=X(J-Q
      X(J)=X(J)+Q
    2 CONTINUE

      DO 3 I=2,N
      II=I
      IF(K.LT.MMM(I)) GO TO 4
```

```
    3   K=K-MMM(I)

    4   K=K+MMM(II)

        K=0

        DO 7 J=1,LX
        IF(K.LT.J) GO TO 5
        HOLD=X(J)
        X(J)=X(K+1)
        X(K+1)=HOLD

    5   DO 6 I=1,N
        II=I
        IF(K.LT.MMM(I)) GO TO 7
    6   K=K-MMM(I)

    7   K=K+MMM(II)

        IF(SIGN.LT.0.0) RETURN

        DO 8 I=1,LX
    8   X(I)=X(I)/FLX

        RETURN
        END


C 'TAMP' CALCULATES THE TIME OF ARRIVAL AND AMPLITUDE OF A
C PARTICULAR PHASE GIVEN ONLY THE RAY CODE
        SUBROUTINE TAMP (ICP1,LAY,ICM,ICN,DREFCO,DTPACO,IIG,THICK,
     CUREFCO,UTRACO,TIME,AF,B,P50,VEL)
        COMPLEX B,CMPLX
        COMPLEX P50(8192)
        DIMENSION I3(16)
        DIMENSION ICN(17)
        DIMENSION  ICM(16)
        DIMENSION DREFCO(16)
        DIMENSION DTRACO(16)
        DIMENSION UREFCO(16)
        DIMENSION UTRACO(16)
        DIMENSION TIME(16)
        DIMENSION THICK(16)
        DIMENSION VEL(15)
   50   FORMAT('AMP =',E15.4)
   51   FORMAT('TIME =',F10.5)
        ICM(ICP1)=ICN(ICP1)
        AMP=1.0
        I=1
        GSPF=0.0
        TIM=0.0
        I3(1)=1

        DO 52 J=2,LAY
```

```
      I3(J)=0
 52   CONTINUE

 53   IF(ICM(I) .GT. 0) GO TO 54
      AMP=AMP*DTRACO(I)
      I=I+1
      I3(I)=I3(I)+1

      GO TO 53

 54   IND50=1
 55   AMP=AMP*DREFCO(I)
      IM1=I-1
      IF(IM1 .EQ. 0) GO TO 100

      DO 56 IB=1,IM1
      IMIB=I-IB
      IZ=IB
      IF(ICN(IMIB) .EQ. I3(IMIB)) GO TO 59
      IIB1=I-IB+1
      AMP =AMP*UTRACO(IIB1)
 56   CONTINUE

      AMP=AMP*UREFCO(IMIB)
      IF(IND50 .EQ. ICM(I)) GO TO 60
      IND50 =IND50+1
      IM1=I-1

      DO 57 IB=1,IM1
      AMP=AMP*DTRACO(IB)
 57   CONTINUE

      DO 58 J=1,I
      I3(J)=I3(J)+1
 58   CONTINUE

      GO TO 55

100   IF(IND50 .EQ. ICM(I)) GO TO 101
      AMP=AMP*UREFCO(1)
      I3(I)=I3(I)+1
      IND50=IND50+1

      GO TO 55

101   IF(I .EQ.ICP1) GO TO 150
      AMP=AMP*UREFCO(1)
      I3(I)=I3(I)+1
      AMP=AMP*DTRACO(1)
      I=I+1
      I3(I)=I3(I)+1

      GO TO 53
```

```
59    IF(IND50 .EQ. ICM(I)) GO TO 140
      IIB1=I-IZ+1
      AMP=AMP*UREFCO(IIB1)
      IND50=IND50+1
      I3(I)=I3(I)+1

      GO TO 55

60    DO 104 I1=1,I
      AMP=AMP*DTRACO(I1)
104   CONTINUE

      IP1=I+1

      DO 105 J=1,IP1
      I3(J)=I3(J)+1
105   CONTINUE

      I=I+1

      GO TO 53

140   IF(I .EQ. ICP1) GO TO 150
      IIB1=I-IZ+1
      AMP=AMP*UREFCO(IIB1)

      DO 141 J=IIB1,I
      AMP=AMP*DTRACO(J)
141   CONTINUE

      IP1=I+1

      DO 142 J=IIB1,IP1
      I3(J)=I3(J)+1
142   CONTINUE

      I=I+1

      GO TO 53

150   DO 151 I=1,ICP1
      AMP=AMP*UTRACO(I)
151   CONTINUE

      IF (IIG .EQ. 0) GO TO 154

      DO 153 I=1,ICP1
      GSPF=GSPF+2*ICN(I)*THICK(I)*VEL(I)/VEL(1)
153   CONTINUE

      AMP=AMP*AF/GSPF

      GO TO 155
```

```
 154    AMP=AMP*AF

 155    DO 152 I=1,ICP1
        TIM=TIM+2*ICN(I)*TIME(I)
  152   CONTINUE

C SPECIFY THE TIME CUTOFF IN THE NEXT STATEMENT
       IF(TIM .GT. 8.) GO TO 257
C       WRITE(6,50) AMP
C       WRITE(6,51) TIM
        B=CMPLX(AMP,TIM)
        CALL RDOFF(B,P50)
 257     RETURN
        END

C 'PSY' PLOTS THE SYNTHETIC SEISMOGRAM
        SUBROUTINE PSY(C,C201,P60)
        DIMENSION C(4602)
        DIMENSION P60(4602)
        DIMENSION C201(4602)
        CALL PLOTS
C MOVE ORIGIN IN TO ALLOW FOR LABELS ETC.
        CALL PLOT(8.0,8.0,-3)
C FOR THE TIME VALUES FORCE 5 INCHES PER SEC.
        XMIN=0.0
        XDELT=.2
C STORE SCALING PARAMETERS FOR THE TIME AXIS
        C201(4601)=XMIN
        C201(4602)=XDELT
C FOR THE AMPLITUDE AXIS WILL FORCE 1./INCH
        YMIN=0.0
        YDELT=.02
C STORE SCALING PARAMETERS FOR AMPLITUDE AXIS
        C(4601)=YMIN
        C(4602)=YDELT
        P60(4601)=YMIN
        P60(4602)=YDELT
C PLOT TIME AXIS
        CALL AXIS(0.0,-1.0,'TIME-SEC',-8,15.0,0.0,0.0,.2)
C PLOT AMPLITUDE AXIS
        CALL AXIS(0.0,-1.0,'AMPLITUDE',9,2.0,90.0,-1.,1.)
C PLOT DATA
        CALL PLOT(0.0,0.0,3)
        DO 1 I=1,200
        XP=(C201(I-C201(4601))/C201(4602)
        TP=C(I)/C(4602)
        CALL PLOT(XP,TP,2)
  1     CONTINUE
        DO 3 I=201,3000
        XP=C201(I)
        TP=C(I)/.0001
        CALL PLOT(XP,TP,2)
  3     CONTINUE
C PLOT SPIKE SYNTHETIC
```

```
C MOVE ORIGIN UP
      CALL PLOT(8.0,16.0,-3)
C PLOT TIME AXIS
      CALL AXIS(0.0,-1.0,'TIME-SEC',-8,15.0,0.0,0.0,.2)
C PLOT AMPLITUDE AXIS
      CALL AXIS(0.0,-1.0,'AMPLITUDE',9,2.0,90.0,-1.,1.)
C PLOT DATA FOR SPIKE SYNTHETIC
      CALL PLOT(0.0,0.0,3)
      DO 2 I=1,200
      XP=(C201(I-C201(4601))/C201(4602)
      TP=P60(I)/P60(4602)
      CALL PLOT(XP,0.0,2)
      CALL PLOT(XP,TP,2)
      CALL PLOT(XP,0.0,2)
 2    CONTINUE
      DO 4 I=201,3000
      XP=C201(I)
      TP=C(I)/.0001
      CALL PLOT(XP,0.,2)
      CALL PLOT(XP,TP,2)
      CALL PLOT(XP,0.,2)
  4   CONTINUE
      CALL PLOT(2.0,2.0,999)
      RETURN
      END


C SUBROUTINE KLAUD GENERATES A NORMALIZED KLAUDER WAVELET
C F1 - THE TERMINAL LOW FREQUENCY
C IZ - THE FREQUENCY BANDWIDTH IN OCTAVES
C T - THE DURATION OF THE SIGNAL IN SECONDS
C THE GENERATION IS DONE USING EQN.7 S.S.C. TRAINING DEPT.
C CATALOG OF KLAUDER WAVELETS
      SUBROUTINE KLAUD (T,F1,Z,VALUE)
C DIMENSION VALUE( ) IN THE FOLLOWING STATEMENT-SEE MAIN
      COMPLEX VALUE(8192), CMPLX
      PI=3.141592
C 'VALUE(301) IN THE NEXT STATEMENT REPRESENTS THE AUTO-
C CORRELATION OF THE SWEEP AT ZERO SHIFT AND SHOULD
C THEORETICALY HAVE A VALUE OF'T', WHERE 'T' IS THE
C TIME LENGTH OF THE SWEEP. IF NORMALIZATION IS
C DESIRED SPECIFY 'COMPLEX(1.0,0.0)' RATHER  THAN
C COMPLEX(T,0.0)IN THE NEXT STATEMENT
      VALUE(301)=CMPLX(1.0,0.0)
      I=301
      SHIFT=.001

      DO 16 J=1,300
      A=PI*F1*SHIFT*((2**Z-1)*(1-SHIFT/T)
      B=PI*F1*SHIFT*((2**Z-1)
      C=PI*F1*SHIFT*((2**Z)+1)
      VALUE(I+J)=CMPLX(1.0*SIN(A)*COS(C)/B,0.0)
      VALUE(I-J)=VALUE(I+J)
```

```
      SHIFT=SHIFT+.001
   16 CONTINUE

      DO 18 I=602,8192
      VALUE(I)=CMPLX(0.0,0.0)
   18 CONTINUE

      RETURN
      END

C SUBROUTINE WAVLT1 STORES IMP. LOG WAVELET
C IN VALUE
      SUBROUTINE WAVLT1(VALUE)
      COMPLEX VALUE(8192),CMPLX
      DIMENSION PALL(97)
   10 FORMAT(5F10.5)
      READ(7,10) (PALL(I),I=1,97)

      DO 15 I=1,97
      VALUE(I)=CMPLX(PALL(I),0.0)
   15 CONTINUE

      DO 20 I=98,8192
      VALUE(I)=CMPLX(0.0,0.0)
   20 CONTINUE

      RETURN
      END



      SUBROUTINE PERMUT(MAC,ICN,ICM,MAXLAY,IND,LAY,DREFCO,
     CDTRACO,IIG,THICK,UREFCO,UTRACO,TIME,AF,IAFG,B,P50,VEL)
      DIMENSION N(16,16)
      DIMENSION ICN(17)
      DIMENSION ICM(16)
      DIMENSION IBEN(16)
      DIMENSION DREFCO(15)
      DIMENSION DTRACO(16)
      DIMENSION THICK(16)
      DIMENSION UREFCO(16)
      DIMENSION UTRACO(16)
      DIMENSION TIME(16)
      DIMENSION VEL(15)
      COMPLEX P50(8192)
      DIMENSION J(16)
      COMPLEX B
      ITT=16-MAC+1

      DO 10 J9=1,16

      DO 9 K=ITT,16
      N(J9,K)=ICN(K-ITT+1)
```

```
    9  CONTINUE

   10  CONTINUE

       CALL DICM(ICN,ICM,MAXLAY,MAC,IND,LAY,DREFCO,
     CDTRACO,IIG,THICK,UREFCO,UTRACO,TIME,AF,IAFG,B,P50,VEL)

       DO 11 I=1,16
       J(I)=I
11     CONTINUE

       GO TO 1000

800    IF(J(J2).LE. J3) GO TO 806
       JM1=J(J2-1

       DO 802 I=J3,JM1
       IF(N(J2,J(J2)).EQ.N(J2,I)) GO TO 1004
802    CONTINUE

806    IF(N(J2,J2).EQ.N(J2,J(J2))) GO TO 1004

       DO 801  I=1,MAC
       ITMAC=16-MAC+I
       N(J3,ITMAC)=N(J2,ITMAC)
801    CONTINUE

       N(J3,J2)=N(J2,J(J2))
       N(J3,J(J2))=N(J2,J2)

       DO 805 I=1,MAC
       IBEN(I)=N(J3,16-MAC+I)
805    CONTINUE

       CALL DICM(IBEN,ICM,MAXLAY,MAC,IND,LAY,DREFCO,
     CDTRACO,IIG,THICK,UREFCO,UTRACO,TIME,AF,IAFG,B,P50,VEL)
       ISEV=16-J1+3

       DO 804 L=ISEV,16

       DO 803  K=1,16
       N(L,K)=N(J3,K)
 803   CONTINUE

804    CONTINUE

1000   IF(N(15,15).EQ.N(15,16)) GO TO 1001
       N(16,15)=N(15,16)
       N(16,16)=N(15,15)

       DO 904 I=1,MAC
       IBEN(I)=N(16,16-MAC+I)
904    CONTINUE
```

```
      CALL DICM(IBEN,ICM,MAXLAY,MAC,IND,LAY,DREFCO,
     CDTRACO,IIG,THICK,UREFCO,UTRACO,TIME,AF,IAFG,B,P50,VEL)

      GO TO 1001

 1004 IFR=J1

      GO TO 1003

 1001 IFR=3

 1003 DO 1010 J1=IFR,16
      IF(MAC.LT.J1) GO TO 1020
      J2=16-J1+1
      J(J2)=J(J2)+1
      J3=16-J1+2

      DO 1002 J4=J3,16
      J(J4)=J4
 1002 CONTINUE

      IF(J(J2).LE.16) GO TO 800
 1010 CONTINUE

 1020 RETURN
      END


C DICM DETERMINES THE VECTOR ICM
      SUBROUTINE DICM(ICN,ICM,MAXLAY,MAC,IND,LAY,DREFCO,DTRACO,
     CIIG,THICK,UREFCO,UTRACO,TIME,AF,IAFG,B,P50,VEL)
      DIMENSION MMN(16)
      DIMENSION MMX(16)
      DIMENSION M(16)
      DIMENSION ICN(17)
      DIMENSION ICM(16)
      DIMENSION DREFCO(16)
      DIMENSION DTRACO(16)
      DIMENSION THICK(16)
      DIMENSION UTRACO(16)
      DIMENSION UREFCO(16)
      DIMENSION TIME(16)
      DIMENSION VEL(15)
      COMPLEX P50(8192)
      COMPLEX B
 23   FORMAT(//,'GROUP NUMBER',T25,I5)
 24   FORMAT('ICN =',T25,10I5)
 25   FORMAT('ICM =',T25,10I5)
 26   FORMAT('N =',F8.3)
      IC=MAC-1

      DO 260 I=1,IC
      MMX(I)=ICN(I-1
```

```
      MMN(I)=MAX0(ICN(I-ICN(I+1),0)
 260  CONTINUE

 264  M(1)=MMN(1)
      I25=MAXLAY-2
 309  I20=1

 310  DO 350 I15=I20,I25
      IF (IC.EQ.I15) GO TO 399
      I30=I15+1
      M(I30)=MMN(I30)
 350  CONTINUE

 399  ICP1=IC+1

 400  DO 450 I=1,IC
      ICM(I)=M(I)
 450  CONTINUE

      IND=IND+1
C     WRITE(6,23) IND
C     WRITE(6,24) (ICN(I),I=1,ICP1)
C     WRITE(6,25) (ICM(I),I=1,IC)
      CALL NODYAN(MAC,ICN,ICM,AF)
C     WRITE(6,26) AF
      IJAF=IFIX(AF)
      IAFG=IAFG+IJAF
      CALL TAMP(MAC,LAY,ICM,ICN,DREFCO,DTRACO,IIG,THICK,UREFCO,
     CUTRACO,TIME,AF,B,P50,VEL)
 331  I200=MAXLAY-2
      MLM1=MAXLAY-1

      DO 520 I100=1,I200
      I201=MLM1-I100
      IF (IC.EQ.I201) GO TO 559
 520  CONTINUE

      M(MLM1)=M(MLM1)+1
      IF(M(MLM1) .LE. MMX(MLM1)) GO TO 400
 559  IF (I201.EQ.1) GO TO1011
 560  M(I201)=M(I201)+1
      IF(M(I201) .LE. MMX(I201)) GO TO 1020
      I201=I201-1

      GO TO 559

1011  M(1)=M(1)+1
      IF(M(1) .LE. MMX(1)) GO TO 309
      IF (M(1) .GT. MMX(1) ) GO TO 650
1020  I20=I201

      GO TO 310

 650  RETURN
```

```
      END
C MAIN FOLLOWS


 500  FORMAT(//,T10,'DOWNWARD REFLECTION COEFFICIENTS',//)
 501  FORMAT(//,T10,'DOWNWARD TRANSMISSION COEFFICIENTS',//)
 502  FORMAT(//,T10,'UPWARD REFLECTION COEFFICIENTS',//)
 503  FORMAT(//,T10,'UPWARD TRANSMISSION COEFFICIENTS',//)
 14   FORMAT(F10.2)
   9  FORMAT(8F10.5)
  10  FORMAT(8F10.5)
  11  FORMAT(8F10.5)
  12  FORMAT(8F10.5)
C DIMENSION VEL(NO. OF VELOCITY VALUES AVAILABLE)
      DIMENSION VEL(15)
C DIMENSION DEPTH(NO. OF VELOCITY VALUES - 1)
      DIMENSION DEPTH(14)
C LAY = NO. OF VELOCITY VALUES - 1
      LAY=14
C THICK(I) CONTAINS THE THICKNESS OF THE (I)TH LAYER
C THE VALUES THICK(I) ARE CALCULATED INTERNALLY GIVEN THE
C DEPTH VALUES
C DIMENSION THICK(NO. OF VELOCITY VALUES - 1)
      DIMENSION THICK(14)
C DREFCO REFERS TO THE DOWNWARD REFLECTION COEFFICIENT
C DIMENSION DREFCO(NO. OF VELOCITY VALUES - 1)
      DIMENSION DREFCO(14)
C DTRACO REFERS TO THE DOWNWARD TRANSMISSION COEFFICIENT
C DIMENSION DTRACO(NO. OF VELOCITY VALUES - 1)
      DIMENSION DTRACO(14)
C UREFCO REFERS TO THE UPWARD REFLECTION COEFFICIENT
C DIMENSION UREFCO(NO. OF VELOCITY VALUES - 1)
      DIMENSION UREFCO(14)
C UTRACO REFERS TO THE UPWARD TRANSMISSION COEFFICIENT
C DIMENSION UTRACO(NO. OF VELOCITY VALUES - 1)
      DIMENSION UTRACO(14)
C TIME(I) CONTAINS THE ONE WAY VERTICAL TRAVEL TIME THROUGH
C LAYER I
C DIMENSION TIME(NO. OF VELOCITY VALUES - 1)
      DIMENSION TIME(14)
C VALUE CONTAINS THE KLAUDER (RIKER) WAVELET PLUS TRAILING
C ZEROS. THE DIMENSION OF VALUE MUST BE GREATER THAN OR
C EQUAL TO THE LENGTH OF THE CONVOLUTION OF THE WAVELET
C WITH THE SPIKE SYNTHETIC, WITH THE ADDED CONDITION THAT
C THE DIMENSION MUST EQUAL 2**N WHERE N IS SOME POSITIVE
C INTERGER.    2**5=32  2**6=64  2**7=128  2**8=256  2**9=512
C 2**10=1024  2**11=2048  2**12=4096  2**13=8192  2**14=16384
      COMPLEX VALUE(8192)
C P50 CONTAINS THE SPIKE SYNTHETIC(OUTPUT FROM RDOFF)
C DIMENSION P50(SAME DIMENSION AS VALUE)
      COMPLEX P50(8192)
C DIMENSION BNUIS(2 * DIMENSION OF VALUE)
      DIMENSION BNUIS(16384)
      EQUIVALENCE(VALUE,BNUIS)
```

```
C C CONTAINS THE Y OR AMPLITUDE VALUES USED IN PLOTING THE
C SYNTHETIC SEISMOGRAM
C DIMENSION C(LENGTH OF THE CONVOLUTION OF THE SPIKE SYNTHETIC
C WITH THE WAVELET  +  2) THE SCALING PARAMETERS ARE STORED IN
C THE LAST TWO POSITIONS
      DIMENSION C(4602)
C C201 CONTAINS THE X OR TIME VALUES USED IN PLOTTING THE SS
C DIMENSION C201(SAME AS DIMENSION OF C)
      DIMENSION C201(4602)
C IN NLOGN THE DATA LENGTH MUST BE EQUAL TO 2**N WHERE N IS
C SOME POSITIVE INTERGER. DIMENSION MMM(LARGEST VALUE OF N TO
C BE PROCESSED)
      DIMENSION MMM(13)
C DIMENSION I3(LAY)
      DIMENSION I3(14)
      DIMENSION IBEN(14)
      DIMENSION P60(4600)
C IF THE GEOMETRICAL SPREADING FACTOR IN AMPLITUDE IS NOT
C DESIRED SPECIFY IIG=O. ANY OTHER VALUE FOR IIG WILL
C INVOKE THE SPREADING CALCULATION. BE CERTAIN TO
C CHANGE THE VERTICAL SCALING PARAMETERS IN THE
C PLOTTING ROUTINE WHEN RUNNING THIS PROGRAM
C WITHOUT GEOMETRICAL SPREADING.
      IIG=1
      UTRACO(1)=1.0
      UREFCO(1)=1.0
C INPUT VELOCITY AND DEPTH VALUES
      READ(8,14)  (DEPTH(I),I=1,LAY)
      JJ50=LAY+1
      READ(10,14)  (VEL(I),I=1,JJ50)
C INSERT DO LOOP PARAMETER ...I=1,DIMENSION OF VALUE

      DO 13 I=1,8192
      P50(I)=CMPLX(0.0,0.0)
   13 CONTINUE

      THICK(1)=DEPTH(1)
      LM1= LAY-1

      DO 8 I=2,LAY
      IM1=I-1
      THICK(I)=DEPTH(I-DEPTH(IM1)
    8 CONTINUE

      DO 3 I=1,LAY
      TIME(I)=THICK(I)/VEL(I)
    3 CONTINUE

      DO 4 I=1,LAY
      IP1=I+1
      DREFCO(I)=(VEL(I-VEL(IP1))/(VEL(I)+VEL(IP1))
   4  CONTINUE

C     WRITE(6,500)
```

```
C       WRITE(6,9)  (DREFCO(I),I=1,LAY)

        DO 5 I=1,LAY
        DTRACO(I)=1+DREFCO(I)
    5   CONTINUE

C       WRITE(6,501)
C       WRITE(6,10)  (DTRACO(I),I=1,LAY)

        DO 6 I=2,LAY
        IM1=I-1
        UREFCO(I)=(VEL(I-VEL(IM1))/(VEL(IM1)+VEL(I))
    6   CONTINUE

C       WRITE(6,502)
C       WRITE(6,11)  (UREFCO(I),I=1,LAY)

        DO 7 I=2,LAY
        UTRACO(I)=1.+UREFCO(I)
   7    CONTINUE

C       WRITE(6,503)
C       WRITE(6,12)  (UTRACO(I),I=1,LAY)
C AFTER EACH RAY CODE IS GENERATED THE AMPLITUDE AND TIME OF
C ARRIVAL ARE CALCULATED AND STORED IN B.
        COMPLEX B
C DIMENSION MMN(MIN(IHMS,LAY))
        DIMENSION MMN(14)
C DIMENSION MMX(SAME AS MMN)
        DIMENSION MMX(14)
C DIMENSION ICN(SAME AS MMN + 1)
        DIMENSION ICN(15)
C DIMENSION ICM(SAME AS MMN)
        DIMENSION ICM(14)
C DIMENSION M(SAME AS MMN)
        DIMENSION M(14)
       MAXLAY=LAY
        IND=0
        IAFG=0
C INPUT IHMS = HALF THE MAX. NO. OF SEGMENTS ALLOWED
        IHMS=14
    21  FORMAT(//,'GROUP NUMBER',T25,I5)
    22  FORMAT(//,T31,'FINISHED',//)
    23  FORMAT(//,'GROUP NUMBER',T25,I5)
    24  FORMAT('ICN =',T25,10I5)
    25  FORMAT('ICM =',T25,10I5)
    26  FORMAT('N =',F10.5)
    27  FORMAT('N =   1')
    50  FORMAT('AMP =',F10.5)
    51  FORMAT('TIME =',F10.5)
    28  FORMAT('ICN =',T25,I5)
   504  FORMAT(/,'TOTAL NUMBER OF PHASES',T30,I10)
```

```
C  PARTITION

 3000   FORMAT('MADE IT')
 4000   FORMAT('PARTITION =',T25,10I5)

        DO 100 N=1,IHMS
        MQ=1
C IN THE NEXT STATEMENT "IF(N.GT.X)..
C TO OBTAIN THE UNRESTRICTED PARTITIONS SET
C X > HMS. FOR SEVERITY LEVEL 1, X=5...FOR
C SEVERITY LEVEL 2, X=4...FOR SEVERITY LEVEL 3
C X=3...FOR SEVERITY LEVEL 4, X=2.
        IF(N.GT.15) GO TO 85
        GO TO 71
C IN THE NEXT STATEMENT "MQ=N-Y"...FOR SEVERITY
C LEVEL 1 SET Y=4...SEVERITY LEVEL 2, Y=3...
C SEVERITY LEVEL 3, Y=2...SEVERITY LEVEL 4, Y=1
 85     MQ=N-2
        MAC=MQ
        GO TO 70
 71     IND=IND+1
        MAC=MQ
        ICN(1)=N
C       WRITE(6,23) IND
C       WRITE(6,28) ICN(1)
C       WRITE(6,27)
        IAFG=IAFG+1
        CALL TAMP(MAC,LAY,ICM,ICN,DREFCO,DTRACO,IIG,THICK,
      CUREFCO,UTRACO,TIME,1.0,B,P50,VEL)

        GO TO 80

   70   MACM1=MAC-1

        DO 72 I=1,MACM1
        ICN(I)=1
   72   CONTINUE

   74   ISUM=ICN(1)
        IF(MACM1.EQ.1)GO TO 78

        DO 76 I=2,MACM1
        ISUM=ISUM+ICN(I)
   76   CONTINUE

   78   ICN(MAC)=N-ISUM
        CALL PERMUT(MAC,ICN,ICM,MAXLAY,IND,LAY,DREFCO,DTRACO,
      CIIG,THICK,UREFCO,UTRACO,TIME,AF,IAFG,B,P50,VEL)
   80   IT=MAC-1
   82   IF(IT.EQ.0) GO TO 88
        IF((ICN(MAC-ICN(IT)).GT.1)GO TO 84
        IT=IT-1

        GO TO 82
```

```
   84   ICRAP=ICN(IT)

        DO 86 I=IT,MACM1
        ICN(I)=ICRAP+1
   86   CONTINUE

        GO TO 74

   88   MAC=MAC+1
        IF(MAC.LE.N)GO TO 70
  100   CONTINUE


 2000   WRITE(6,504)  IAFG
        WRITE(6,22)
C GENERATE THE TIME AXIS FOR PLOTTING
        C201(1)=0.
C INSERT DO LOOP PARAMATER. I=2,LENGTH OF TRACE
C IN SECONDS X 1000

        DO 2001 I=1,4599
        C201(I+1)=I/1000.
 2001   CONTINUE

C       WRITE(7,2004)  (P50(I),I=1,8192)
C       FORMAT(2F10.5)
C SPECIFY THE KLAUDER WAVELET PARAMETERS
C       CALL KLAUD(7.,6.,4.0,VALUE)
        CALL WAVLT1(VALUE)

        DO 2004 I=1,4602
        P60(I)=REAL(P50(I))
 2004   CONTINUE

        WRITE(11) P60
C IN THE NEXT TWO STATEMENTS THE FIRST PARAMETER OF
C NLOGN MUST EQUAL THE DIMENSION OF MMM( )
        CALL NLOGN(13,VALUE,-1.0)
        CALL NLOGN(13,P50,-1.0)

        DO 2002 I=1,8192
        VALUE(I)=VALUE(I)*P50(I)
 2002   CONTINUE

C INSERT 1ST NLOGN PARAMETER
        CALL NLOGN(13,VALUE,1.0)
C INSERT DO LOOP PARAMETER. I=1,LENGTH OF TRACE
C IN SECONDS X 1000

        DO 2003 I=1,4600
        C(I)=BNUIS(2*I-1)
 2003   CONTINUE
```